박 사 학 위 논 문
Ph.D. Dissertation

디퓨전 모델 기반 역문제에서 사후 샘플링의 실용적 근사 방법

Practical approximations of posterior sampling in diffusion model-based inverse problems

2025

정 형 진 (鄭 衡 辰 Chung, Hyungjin)

한 국 과 학 기 술 원

Korea Advanced Institute of Science and Technology

박사학위논문

디퓨전 모델 기반 역문제에서 사후 샘플링의 실용적 근사 방법

2025

정형진

한국과학기술원

바이오및뇌공학과

디퓨전 모델 기반 역문제에서 사후 샘플링의 실용적 근사 방법

정 형 진

위 논문은 한국과학기술원 박사학위논문으로
학위논문 심사위원회의 심사를 통과하였음

2024년 12월 5일

심사위원장 예 종 철 (인)
심 사 위 원 전 세 영 (인)
심 사 위 원 신 진 우 (인)
심 사 위 원 성 민 혁 (인)
심 사 위 원 이 주 호 (인)

Practical approximations of posterior sampling in diffusion model-based inverse problems

Hyungjin Chung

Advisor: Jong Chul Ye

A dissertation submitted to the faculty of
Korea Advanced Institute of Science and Technology in
partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Bio and Brain Engineering

Daejeon, Korea
December 5, 2024

Approved by

Jong Chul Ye
Kim Jae Chul Graduate School of AI

The study was conducted in accordance with Code of Research Ethics[1].

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of Korea Advanced Institute of Science and Technology, hereby declare that I have not committed any act that may damage the credibility of my research. This includes, but is not limited to, falsification, thesis written by someone else, distortion of research findings, and plagiarism. I confirm that my thesis contains honest conclusions based on my own careful research under the guidance of my advisor.

DBIS 정형진. 디퓨전 모델 기반 역문제에서 사후 샘플링의 실용적 근사 방법. 바이오및뇌공학과 . 2025년. 106+iv 쪽. 지도교수: 예종철. (영문 논문)
Hyungjin Chung. Practical approximations of posterior sampling in diffusion model-based inverse problems. Department of Bio and Brain Engineering . 2025. 106+iv pages. Advisor: Jong Chul Ye. (Text in English)

초 록

본 논문은 생성 프라이어(generative prior)로서의 디퓨전 모델을 활용한 역문제 해결의 새로운 접근 방식을 제안한다. 전통적인 딥러닝 접근법은 주로 최소 제곱 오차(MMSE) 추정값을 도출하기 위해 지도 학습을 사용하지만, 이 접근 방식은 후방 분포로부터의 샘플링의 이점을 놓치는 경우가 많다. 반면, 본 연구의 접근법은 후방 분포로부터의 샘플링을 가능하게 하여 향상된 지각 품질과 불확실성 정량화를 제공한다. 우리는 광범위한 역문제를 해결하기 위한 실용적인 후방 샘플링 근사 기법으로서, 확장성 있는 Diffusion Posterior Sampling(DPS) 프레임워크를 소개한다. 또한, 3D 대규모 이미지 복원과 같은 복잡한 문제에 DPS를 적용하기 위해 디퓨전 모델의 기하학적 특성을 활용한 Decomposed Diffusion Sampler를 제안한다. 더 나아가, 데이터 부족 상황에서의 프라이어 불일치를 완화하기 위해 Deep Diffusion Image Prior라는 온라인 적응 방법을 도입하여 학습 데이터 없이도 효율적인 샘플링이 가능함을 시연한다. 마지막으로, 텍스트 기반 메타데이터를 추가 정보로 활용하여 신호 복원을 향상시키거나 특정 모드를 향해 샘플링을 유도할 수 있는 제어 메커니즘으로서의 가능성을 탐구한다. 본 논문에서 제안한 방법들은 디퓨전 모델 기반 역문제 해결에 대한 견고한 이론적 원칙을 기반으로 하며, 실질적인 구현 전략을 통해 실제 적용 가능성을 확보한다.

핵 심 낱 말 확산 모델, 역문제, 사후 샘플링

Abstract

This thesis presents an innovative approach to inverse problem solving by employing diffusion models as generative priors. Traditional deep learning approaches predominantly utilize supervised learning to obtain the minimum mean squared error estimate, often overlooking the advantages of sampling from the posterior distribution. By contrast, our approach enables posterior sampling, which significantly enhances perceptual quality and facilitates uncertainty quantification. We introduce a robust framework, Diffusion Posterior Sampling (DPS), as a practical approximation for posterior sampling that can address a broad spectrum of inverse problems, including non-linear and blind scenarios. To further augment DPS's applicability to complex tasks, we extend its usage to large-scale 3D image reconstruction and propose the Decomposed Diffusion Sampler, a method that leverages the inherent geometric properties of diffusion models. To mitigate prior mismatch, we also introduce the Deep Diffusion Image Prior, an online adaptation technique that enables effective sampling in cases where gold-standard data for training a suitable prior is unavailable. Finally, we explore the incorporation of text-based metadata as supplementary information that can either enhance signal recovery or act as a control mechanism, steering the sampling process towards specific outcomes. The methods developed in this thesis are underpinned by a rigorous theoretical framework for diffusion model-based solvers tailored to inverse problems, complemented by practical implementation strategies to facilitate real-world applicability.

Keywords Diffusion Models, Inverse Problems, Posterior Sampling

Contents

List of Tables

List of Figures

Chapter 1. Introduction

Inverse problems are ubiquitous in science and engineering. It is seldom the case where you have full information about the (sub-)problem that you are aiming to solve. In the simplest case, your measurement will be noisy, and one would have to apply some denoising to correct for this. In more challenging cases, some measurements could be completely missing, and one would have to fill in the missing values. In either case, we are given an *ill-posed* inverse problem with infinitely many *feasible* solutions. Note that feasibility does not imply that the achieved solution is *good*. For instance, consider the case of image inpainting, where the central part of the image is unknown. While filling in all zeros in the missing region would result in a perfectly feasible solution, by no means would it be a good solution, as such an image would be highly unrealistic. In this regard, a natural question arises—what is a *good* solution? A natural answer would be a solution that lies in the natural data manifold. In a probabilistic sense[1], these solutions would correspond to the ones having high prior probability. The problem now boils down to retrieving a solution that has both high prior probability (good solution) and high likelihood (feasible solution).

Classically, the prior probability was hand-crafted by inspection. For instance, one of the most widely used priors was sparse solutions in some domain [40]. Unfortunately, hand-crafted priors are often not expressive enough to describe the true prior. As deep learning gained popularity, the standard method for solving inverse problems shifted to supervised learning, where a large collection of paired ground truth and measurement data were collected to train a network to directly invert the measurement process. The specialist models are trained to minimize the empirical risk [164] of the training set. When the risk is the minimum squared error (MSE) between the output of the network and the ground truth target (which is often the case), then the network is trained to output minimum mean squared error (MMSE) estimates. While supervised learning-based methods were superior to optimization methods that were based on hand-crafted priors, they have several limitations. First, specialist models lack generalization, and one needs to train a specific model for all the different problems at hand. Second, the learning part (of the prior) is only done implicitly, making it hard to decipher what the model has learned. Third, the nature of MMSE training yields *blurry* solutions [9].

In recent years, we experienced exponential development of deep generative models [80, 62, 156], and among them, especially diffusion models [62, 156] became predominant in the vision community. Training a diffusion model amounts to fitting a neural network that approximates the gradient of the log data (i.e. prior) distribution, called the *score function*, in multiple levels of granularity. This, in turn, brings access to the un-normalized density. Due to the well-established Tweedie's formula [43], learning the score function is equivalent to training an MMSE denoiser,[2] but now in multiple different noise levels.

In this thesis, we will explore how we can leverage powerful diffusion models as general priors for solving diverse inverse problems in an unsupervised fashion. The resulting methods will enable acquiring stochastic samples that approximate posterior sampling and will have several properties that were not possible from previous approaches. For one, the prior will be much more expressive compared to the traditional hand-crafted priors. Moreover, the samples acquired through sampling will have high perceptual quality, avoiding the downsides of the typical MMSE estimates from supervised learning. Throughout the thesis, we often use the abbreviation DIS, standing for diffusion model-based inverse problem sovlers.

The thesis will be structured as follows. In Chapter 2 we will review the basics of inverse problems and

[1] Covered in more detail in Section 2.1

[2] See Appendix A.1 for formal treatment.

diffusion models. In Chapter 3, we introduce our key method, diffusion posterior sampling (DPS), and build upon it to tackle more and more challenging problems along the following sections. We review how we can solve general noisy, non-linear, and even blind inverse problems leveraging the power of diffusion prior. In Chapter 4, we propose ways to solve 3D inverse problems and accelerate the inverse problem solver so that it applies to practical large-scale inverse problems arising in medical imaging. In Chapter 5, we go a step further, proposing a way to adapt the diffusion prior to OOD measurements, useful for cases in scientific imaging where training with high-quality data is impossible. Finally, in Chapter 6, we study text-driven inverse problems, where we show that additionally incorporating text information can improve the performance of inverse problem solving, and lessons from DIS literature naturally applies to text-to-image synthesis tasks.

Chapter 2. Background

2.1 Inverse problems

2.1.1 Problem setting

We consider the following problem, which, despite its simplicity, can be used to model the measurement process of many different physical measurement processes

$$\boldsymbol{y} = \mathcal{A}(\boldsymbol{x}) + \boldsymbol{n}, \quad \boldsymbol{y} \in \mathbb{R}^m, \quad \boldsymbol{x} \in \mathbb{R}^n, \quad \mathcal{A} : \mathbb{R}^n \mapsto \mathbb{R}^m, \quad \boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \sigma_y^2 \boldsymbol{I}) \tag{2.1}$$

where $m < n$, making the problem ill-posed. i.e. There exist infinitely many solutions. The problem is to reconstruct a clean signal $\boldsymbol{x}$ from the deficient and noisy measurement $\boldsymbol{y}$. Often, measurement systems are linear, or at least well-approximated by a linear matrix (e.g. medical imaging, image degradation process). In such case, the problem simplifies to $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. In a probabilistic framework, Eq. (2.1) is often interpreted as the likelihood $p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}|\mathcal{A}(\boldsymbol{x}), \sigma_y^2 \boldsymbol{I})$. One can try to minimize the negative log likelihood to find an $\boldsymbol{x}$ that meets the data constraints

$$-\log p(\boldsymbol{y}|\boldsymbol{x}) = \frac{\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2}{2\sigma_y^2} + \text{constant} \tag{2.2}$$

Note that the problem statement in Eq. (2.1) considers the case where the measurement noise is Gaussian, a signal-independent noise. Often, especially when photon counts from light are what the detector is measuring, the noise is better modeled by a signal-dependent Poisson noise process $\boldsymbol{y} = \mathcal{P}(\boldsymbol{A}\boldsymbol{x})$. The likelihood function then reads

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_i \frac{e^{-(\boldsymbol{A}\boldsymbol{x})_i} (\boldsymbol{A}\boldsymbol{x})_i^{\boldsymbol{y}_i}}{\boldsymbol{y}_i!}, \tag{2.3}$$

where $(\cdot)_i$ represents the i-th component of a vector. The measurement fidelity term reads

$$-\log p(\boldsymbol{y}|\boldsymbol{x}) = -\boldsymbol{y}^\top \log(\boldsymbol{A}\boldsymbol{x}) + \boldsymbol{1}^\top \boldsymbol{A}\boldsymbol{x} + \text{constant} \tag{2.4}$$

While there exists exact formulations Eq. (2.3) and Eq. (2.4), it is known that in practice, Gaussian approximation of the Poisson measurement model yields more robust results. Specifically, the Gaussian approximation for a single pixel reads

$$\boldsymbol{y}_i \approx \boldsymbol{x}_i + \boldsymbol{n}_i, \quad \boldsymbol{n}_i \sim \mathcal{N}(0, \sigma_0^2 \boldsymbol{y}_i) \tag{2.5}$$

With this approximation, the fidelity term leads to least squares similar to the Gaussian case

$$-\log p(\boldsymbol{y}|\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_{\boldsymbol{\Lambda}}^2, \quad [\boldsymbol{\Lambda}]_{ii} := 1/2\sigma_0^2 \boldsymbol{y}_i \tag{2.6}$$

where $\|\boldsymbol{a}\|_{\boldsymbol{\Lambda}}^2 := \boldsymbol{a}^\top \boldsymbol{\Lambda} \boldsymbol{a}$.

2.1.2 Supervised learning

One canonical way to solve inverse problems is by learning a direct inversion of the operator $\mathcal{A}$ when we have access to a large amount of paired training data $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N$. In this case, we can train a direct mapping G_θ which produces the estimate of $\boldsymbol{x}$ with a single forward pass through the network. The training process reads

$$\theta^* = \arg\min_\theta \sum_{i=1}^{N} d(\hat{\boldsymbol{x}}_i, \boldsymbol{x}_i) + \lambda R(\hat{\boldsymbol{x}}_i), \quad \hat{\boldsymbol{x}}_i := G_\theta(\boldsymbol{y}_i), \tag{2.7}$$

where $d(\cdot, \cdot)$ is a metric that computes the distance between the two arguments, and we optionally have an additional regularizer $R(\cdot)$ on top of the standard paired loss. The distance measure is often ℓ_2 but not necessarily so, where some other popular choices being ℓ_1 or the perceptual distance [180]. However, regardless of the choice, using only the supervised loss without additional regularization leads to the regression-to-the-mean effect [99], with blurry outputs as a consequence. To counteract this effect, GAN loss has been popular in the place of $R(\cdot)$, which further enforces some generative prior [99, 169].

While the supervised learning scheme has been studied across widely different areas of inverse problems over the years [99, 74, 119], a critical downside exists. The model learns to invert the measurement process $\mathcal{A}$ only seen during the training and does not generalize to other processes that are not seen. Consequently, one often needs to train a *specialist* model for each degradation. This is in stark contrast to inverse problem solvers that will leverage a *general* generative prior for all the different problems at hand, without the need to train the model for a specific task, as will be seen in further detail in Sec. 2.1.3.

2.1.3 Bayesian inference

In a probabilistic sense, this can be represented as the following *likelihood* model

$$p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}; \boldsymbol{A}\boldsymbol{x}, \sigma_y^2 \boldsymbol{I}) = Z \exp\left(-\frac{\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2}{2\sigma_y^2}\right), \tag{2.8}$$

where Z is a normalizing constant. On the other hand, we are interested in the *posterior* distribution, which is achieved by Bayes rule

$$\overbrace{p(\boldsymbol{x}|\boldsymbol{y})}^{\text{posterior}} \propto \overbrace{p(\boldsymbol{x})}^{\text{prior}} \overbrace{p(\boldsymbol{y}|\boldsymbol{x})}^{\text{likelihood Eq. (2.8)}}. \tag{2.9}$$

To access the posterior, one needs to define a suitable prior $p(\boldsymbol{x})$, which can be thought of as the **naturalness** of the signal.

Intuitive meaning of prior and the likelihood

High posterior probability requires high prior probability conjugated with a high likelihood value. A high prior probability means that the image is realistic, while a low prior probability means that the image is unrealistic. However, in order to also achieve a high likelihood, $\boldsymbol{x}$ should not be just *any* realistic image. It should adhere to the measurement information contained in $\boldsymbol{y}$. This is often denoted as data consistency, measurement consistency, fidelity, etc.

The advances in inverse problem-solving can be attributed to the advancements in devising a better prior. Traditional methods used hand-crafted priors: total variation [10, 44], sparsity [40, 112] are two of the most widely

used priors that are used in the context of medical imaging. Once we define a prior, we can choose to either find $\boldsymbol{x}$ that maximizes the posterior (i.e. maximum a posteriori; MAP), or to sample from the posterior (i.e. posterior sampling). For instance, performing MAP can be done in the following way. From Eq. (2.9), we can take the log on both sides to have

$$\log p(\boldsymbol{x}|\boldsymbol{y}) = \log p(\boldsymbol{x}) + \log p(\boldsymbol{y}|\boldsymbol{x}) - \log p(\boldsymbol{y}). \tag{2.10}$$

We can equivalently minimize the negative log posterior, leading to

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} - \log p(\boldsymbol{x}) - \log p(\boldsymbol{y}|\boldsymbol{x}) \tag{2.11}$$

$$= \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \frac{1}{2\sigma_y^2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2, \tag{2.12}$$

where $f(\boldsymbol{x}) = \exp(p(\boldsymbol{x}))$ defines our implicit prior. For instance, taking $f(\boldsymbol{x}) := \|\boldsymbol{x}\|_1$ regularizes so that the reconstructed signal is sparse and setting $f(\boldsymbol{x}) := \|\boldsymbol{D}\boldsymbol{x}\|_1$ where $\boldsymbol{D}$ is the finite difference operator regularizes so that the signal is smooth.

Up until now, we discussed point estimates that aim to recover a single reconstruction by for example minimizing the average reconstruction error (i.e., MMSE) or by finding the most probable reconstruction through Maximum a Posteriori estimate (MAP), i.e., finding the $\boldsymbol{x}$ that *maximizes* $p(\boldsymbol{x}|\boldsymbol{y})$. An alternative formulation is to cast the problem as one of sampling, where we are interested in finding samples from the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$. Algorithmically, to solve the former problem we typically use (some variation of) Gradient Descent and to solve the latter (some variation of) Langevin Dynamics. Either way, one needs to compute the gradient of the conditional log-likelihood, i.e. $\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}|\boldsymbol{y})$. A simple Bayes Rule reveals that:

$$\underbrace{\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}|\boldsymbol{y})}_{\text{conditional score}} = \underbrace{\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})}_{\text{unconditional score}} + \underbrace{\nabla_{\boldsymbol{x}} \log p(\boldsymbol{y}|\boldsymbol{x})}_{\text{measurements matching term}} . \tag{2.13}$$

2.2 Diffusion models

2.2.1 Score perspective

Consider the following continuous diffusion process $\boldsymbol{x}(t), t \in [0, T]$ with $\boldsymbol{x}(t) \in \mathbb{R}^d$ [156]. We set $\boldsymbol{x}(0) \sim p_0(\boldsymbol{x})$, where $p_0 = p_{\text{data}}$ as our initial data distribution, and $\boldsymbol{x}(T) \sim p_T$, where p_T is a reference distribution that we can sample from. The forward noising process from $t = 0 \to T$ can be defined by the following Itô stochastic differential equation:

$$d\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}, t)dt + g(t)d\boldsymbol{w}, \quad \boldsymbol{f} : \mathbb{R}^d \times \mathbb{R} \mapsto \mathbb{R}^d, \; g : \mathbb{R} \mapsto \mathbb{R}, \tag{2.14}$$

where $\boldsymbol{f}$ is the drift function of $\boldsymbol{x}(t)$, g is the diffusion coefficient coupled with the standard d-dimensional Brownian motion $\boldsymbol{w} \in \mathbb{R}^d$. By properly choosing $\boldsymbol{f}, g$, one can asymptotically approach the Gaussian distribution as $t \to T$. When the drift function $\boldsymbol{f}$ is taken to be an affine function of $\boldsymbol{x}$, i.e. $\boldsymbol{f}(\boldsymbol{x}, t) = f(t)\boldsymbol{x}$, then the perturbation kernel $p(\boldsymbol{x}(t)|\boldsymbol{x}(0))$ is always Gaussian, where the parameters can be calculated in closed-form. Hence, perturbing the data with the perturbation kernel $p(\boldsymbol{x}(t)|\boldsymbol{x}(0))$ can be done without running the forward SDE. Owing to this property, one never *gradually* adds noise to data when training a diffusion model.

For given forward SDE in Eq. (2.14), it can be shown that there exists a reverse-time SDE running back-

wards [156, 69, 2]:

$$d\boldsymbol{x} = [\boldsymbol{f}(\boldsymbol{x}, t) - g(t)^2 \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})]dt + g(t)d\bar{\boldsymbol{w}} \tag{2.15}$$

where dt is the infinitesimal *negative* time step, and $\bar{\boldsymbol{w}}$ is the standard Brownian motion running backwards. Running the reverse diffusion in Eq. (2.15) by sampling a random gaussian noise as an initial value would lead to sampling from $p_0(\boldsymbol{x})$. In order to do so, it is clear that we need access to the time-conditional score function $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$, which corresponds to the score function of the smoothed data distribution that is convolved with a Gaussian kernel.

An interesting fact is that there exists a corresponding deterministic ODE to Eq. (2.15), which reads

$$d\boldsymbol{x} = [\underbrace{\boldsymbol{f}(\boldsymbol{x}, t) - g(t)^2 \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})}_{=:\tilde{\boldsymbol{f}}_\theta(\boldsymbol{x}, t)}]dt. \tag{2.16}$$

The ODE in Eq. (2.16) is called probability-flow ODE (PF-ODE). While Eq. (2.15) and Eq. (2.16) recover the same law $p_t(\boldsymbol{x})$, PF-ODE has several intriguing properties. First, diffusion models can now be seen as a type of continuous normalizing flows (CNF), by considering the network as $\tilde{\boldsymbol{f}}_\theta$, leading to tractable likelihood computation. Second, ODE solvers are typically more well-behaved compared to SDE solvers. Solving the PF-ODE instead of the reverse SDE leads to faster sampling.

One can train a neural network to approximate the actual score function via a procedure called score matching [154, 156] to estimate $\boldsymbol{s}_\theta(\boldsymbol{x}, t) \approx \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$, and plug it into Eq. (2.15). However, it is known that using explicit or implicit score matching is hardly scalable due to the instability and the compute requirements. To circumvent technical difficulties, denoising score matching (DSM) is used

$$\theta^* = \arg\min_\theta \mathbb{E}_{t\sim\mathrm{Unif}(0,T), \boldsymbol{x}_t\sim p(\boldsymbol{x}_t|\boldsymbol{x}_0), \boldsymbol{x}_0\sim p(\boldsymbol{x}_0)} \left[\|\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{x}_0)\|_2^2\right]. \tag{2.17}$$

It should be noted that DSM, as the name implies, is equivalent to training a denoising autoencoder (DAE) on multiple noise levels, determined by an additional input t. Concretely, consider the simplest forward perturbation kernel $p(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{x}_0, t^2\boldsymbol{I})$[1]. Then, by setting a denoiser parametrization $D_\theta(\boldsymbol{x}_t, t) \triangleq -\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)/t^2$, it is easy to see that Eq. (2.17) can be rewritten as

$$\theta^* = \arg\min_\theta \mathbb{E}_{t\sim\mathrm{Unif}(0,T), \boldsymbol{x}_t\sim p(\boldsymbol{x}_t|\boldsymbol{x}_0), \boldsymbol{x}_0\sim p(\boldsymbol{x}_0)} \left[t\|D_\theta(\boldsymbol{x}_t, t) - \boldsymbol{x}_0\|_2^2\right]. \tag{2.18}$$

The equivalence between Eq. (2.17) and Eq. (2.18) is also related to Tweedie's theorem [43, 87]

Theorem 1 (Tweedie's theorem)**.** *Given a Gaussian perturbation kernel $p(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; s_t\boldsymbol{x}_0, \sigma_t^2\boldsymbol{I})$, the posterior mean is given by*

$$\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t] = \frac{1}{s_t}(\boldsymbol{x}_t + \sigma_t^2 \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)) \tag{2.19}$$

In other words, the parametrization in Eq. (2.18) is a way of directly estimating the posterior mean $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$. Regardless of the parametrization and thanks to 1, diffusion models can be seen as having two dual representations: the noisy variable $\boldsymbol{x}_t$ that evolves with the reverse SDE in Eq. (2.15), and the posterior mean $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$, which is implicitly given by the Tweedie's theorem, and can be thought of as the end point of the trajectory when taking a tangent direction to the current step.

[1]This choice is called the variance exploding (VE) diffusion, as the signal is kept the same throughout the diffusion process, but buried under exploding noise.

2.2.2 Variational perspective

Parallel to the development of the score-based perspective on diffusion models, a variational perspective was also developed [148, 62], which now links diffusion models to VAEs [89]. Specifically, under this perspective, diffusion models are a hierarchical latent variable model called denoising diffusion probabilistic models (DDPM)

$$p_\theta(\boldsymbol{x}_0) = \int p_\theta(\boldsymbol{x}_T) \prod_{t=1}^{T} p_\theta^{(t)}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\, d\boldsymbol{x}_{1:T}, \tag{2.20}$$

where $\boldsymbol{x}_{\{1,\dots,T\}} \in \mathbb{R}^d$. The neural network that models p_θ is then trained by minimizing the evidence lower bound (ELBO)

$$\mathbb{E}[-\log p_\theta(\boldsymbol{x}_0)] \leq \mathbb{E}_q \left[-\log \frac{p_\theta(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \right] = \mathbb{E}_q \left[-\log p(\boldsymbol{x}_T) - \sum_{t\geq 1} \log \frac{p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})} \right] \tag{2.21}$$

where the inference distribution q is defined by the Markovian forward conditional densities

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t|\sqrt{\beta_t}\boldsymbol{x}_{t-1}, (1-\beta_t)I), \tag{2.22}$$

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t|\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)I). \tag{2.23}$$

Here, the noise schedule β_t is an increasing sequence of t, with $\bar{\alpha}_t := \prod_{i=1}^{t} \alpha_t$, $\alpha_t := 1 - \beta_t$. The noise schedule is chosen such that the signal coefficient $\sqrt{\bar{\alpha}_t}$ is sufficiently close to 0 as $t \to T$, which in turn ensures that the noise coefficient $1 - \bar{\alpha}_t$ is sufficiently close to 1, approaching the standard normal distribution. Unlike the choice of VE diffusion discussed in Sec. 2.2.1, the choice made here is called variance preserving (VP). Interestingly, the discrete VP setup in Eq. (2.22), when pushed to the continuous counterpart by setting the number of discretization steps to $N \to \infty$, leads to the following SDE

$$d\boldsymbol{x} = -\frac{1}{2}\beta_t \boldsymbol{x}\, dt + \sqrt{\beta_t} d\boldsymbol{w}. \tag{2.24}$$

Minimizing the ELBO objective in Eq. (2.21) essentially leads to the following optimization problem

$$\min_\theta \mathbb{E}_q \left[\sum_{t>1} D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)) \right]. \tag{2.25}$$

The KL minimization problem in Eq. (2.25) is tractable as both distributions are Gaussians. For the first term, this comes from Bayes rule and the Markov property

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0) \frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} = \mathcal{N}(\boldsymbol{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\boldsymbol{x}_t, \boldsymbol{x}_0), \tilde{\beta}_t \boldsymbol{I}), \tag{2.26}$$

$$\text{where} \quad \tilde{\boldsymbol{\mu}}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t,\ \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \tag{2.27}$$

For the second term, the reverse distribution is Gaussian as we are considering small perturbations for a single step of forward diffusion [62]. A typical parametrization is to set

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \tilde{\beta}\boldsymbol{I}), \tag{2.28}$$

$$\text{where} \quad \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) \right). \tag{2.29}$$

Under this choice, the ELBO objective in Eq. (2.21) can be simplified to the epsilon-matching objective by ignoring the time-dependent weighting factors

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{\boldsymbol{x}_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}_0), \boldsymbol{x}_0 \sim p_{\text{data}}(\boldsymbol{x}_0), \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})} \left[\|\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{\epsilon}\|_2^2 \right]. \tag{2.30}$$

Epsilon matching is equivalent to DSM/DAE objective up to a constant with different parametrization. Given the equivalence of the forward noising distribution in Eq. (2.24) and the learning objective in Eq. (2.17),Eq. (2.30), it can be seen that the two perspectives essentially lead to the same model.

Inference can be done by plugging in the trained $\boldsymbol{\epsilon}_\theta$ to estimate the mean of $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$, leading to the following iteration

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) \right) + \tilde{\beta}_t \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}). \tag{2.31}$$

Notice that similar to the reverse SDE in Eq. (2.15), we add stochastic noise in every iteration during DDPM sampling, leading to slower inference. A canonical way to avoid this, similar to the transition to the PF-ODE, can be done by denoising diffusion implicit models (DDIM) [156], where another inference distribution is introduced

$$q_\eta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0 + \sqrt{1 - \bar{\alpha}_{t-1} - \eta\tilde{\beta}_t^2} \frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}}, \eta\tilde{\beta}_t^2 \boldsymbol{I}), \tag{2.32}$$

where $\eta \in [0, 1]$. By setting $\eta = 1.0$, we recover the original DDPM sampling with maximal stochasticity. By setting $\eta = 0.0$, we achieve a deterministic sampler, which can be shown to be equivalent to the VE PF-ODE [156]. Using smaller values of η leads to better results when the aim is to reduce the number of function evaluations (NFE).

2.2.3 Latent Diffusion

Diffusion models are compute-heavy. This is not only because diffusion models require sequentially querying diffusion models to numerically solve the generative SDE/ODE, but also because the *latent* $\boldsymbol{x}_t$ has the same dimension as the original signal $\boldsymbol{x}_0$. This makes directly scaling diffusion models to high-dimensional signals hard, requiring special treatments to achieve decent results [64, 35]. Also, this is different from most other generative models, where the dimensionality of the latent is much smaller than the signal. This can be especially troubling when one considers the manifold hypothesis, which states that the manifold in which the signal resides, is a low-dimensional space. To mitigate these drawbacks, diffusion models in the latent space were proposed [163, 135].

The construction of the diffusion trajectory is identical to the diffusion in the pixel space. In the first stage of training latent diffusion models (LDMs), only the VAE is trained to compress the signal into a compact representation $\boldsymbol{z} \in \mathbb{R}^k$ with $k < d$

$$\boldsymbol{x} = \mathcal{D}_{\boldsymbol{\varphi}}(\boldsymbol{z}), \quad \text{where} \quad \boldsymbol{z} = \mathcal{E}_{\boldsymbol{\phi}}(\boldsymbol{x}) := \mathcal{E}_{\boldsymbol{\phi}}^{\boldsymbol{\mu}}(\boldsymbol{x}) + \mathcal{E}_{\boldsymbol{\phi}}^{\boldsymbol{\sigma}}(\boldsymbol{x}) \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I}), \tag{2.33}$$

where $\mathcal{E}_\phi$ is the encoder, and $\mathcal{D}_\varphi$ is the decoder. In the second stage, using a pre-trained encoder of the VAE, the diffusion model is trained. In LDMs, a conditioning scheme was also introduced, where the network takes in another input $\boldsymbol{c}$ through cross attention [165], leading to the following training scheme

$$\min_{\theta} \mathbb{E}_{\boldsymbol{z}_t \sim p(\boldsymbol{z}_t|\boldsymbol{z}_0), \boldsymbol{z}_0 \sim p(\mathcal{E}(\boldsymbol{x}_0)), (\boldsymbol{x}_0, \boldsymbol{c}) \sim p(\boldsymbol{x}_0, \boldsymbol{c}) \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})} \left[\|\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t; \boldsymbol{c}) - \boldsymbol{\epsilon}\|_2^2 \right]. \tag{2.34}$$

Setting $\boldsymbol{c}$ to be the text embedding of the pre-trained CLIP [130] encoder, a text-to-image (T2I) diffusion model was constructed. Later, scaling the compute and data led to the popular stable diffusion (SD) [135, 127, 45].

Chapter 3. Diffusion Posterior Sampling

In this chapter, we start by studying the most general method of inverse problem solving with diffusion, which we call diffusion posterior sampling (DPS), by studying the use of Bayesian inference in diffusion models.

3.1 Diffusion models for inverse problems: Basics

For various scientific problems, we have a partial measurement $\boldsymbol{y}$ that is derived from $\boldsymbol{x}$. When the mapping $\boldsymbol{x} \mapsto \boldsymbol{y}$ is many-to-one, we arrive at an ill-posed inverse problem, where we cannot exactly retrieve $\boldsymbol{x}$. In the Bayesian framework, one utilizes $p(\boldsymbol{x})$ as the *prior*, and samples from the *posterior* $p(\boldsymbol{x}|\boldsymbol{y})$, where the relationship is formally established with the Bayes' rule: $p(\boldsymbol{x}|\boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})/p(\boldsymbol{y})$. Leveraging the diffusion model as the prior, it is straightforward to modify Eq. (2.15) to arrive at the reverse diffusion sampler for sampling from the posterior distribution:

$$d\boldsymbol{x} = \left[-\boldsymbol{f}(\boldsymbol{x}, t) - g(t)^2(\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) + \nabla_{x_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t))\right] dt + g(t)\, d\bar{\boldsymbol{w}}, \tag{3.1}$$

where we have used the fact that

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) = \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t). \tag{3.2}$$

In Eq. (3.1), we have two terms that should be computed: the score function $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$, and the likelihood $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t)$. To compute the former term involving $p_t(\boldsymbol{x})$, we can simply use the pre-trained score function $\boldsymbol{s}_{\theta^*}$. However, the latter term is hard to acquire in closed form due to the dependence on the time t, as there only exists explicit dependence between $\boldsymbol{y}$ and $\boldsymbol{x}_0$.

3.2 DPS: General noisy inverse problem solver

3.2.1 Approximation of the likelihood

Recall that no analytical formulation for $p(\boldsymbol{y}|\boldsymbol{x}_t)$ exists. In order to exploit the measurement model $p(\boldsymbol{y}|\boldsymbol{x}_0)$, we factorize $p(\boldsymbol{y}|\boldsymbol{x}_t)$ as follows:

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{x}_t) &= \int p(\boldsymbol{y}|\boldsymbol{x}_0, \boldsymbol{x}_t) p(\boldsymbol{x}_0|\boldsymbol{x}_t) d\boldsymbol{x}_0 \\ &= \int p(\boldsymbol{y}|\boldsymbol{x}_0) p(\boldsymbol{x}_0|\boldsymbol{x}_t) d\boldsymbol{x}_0, \end{aligned} \tag{3.3}$$

where the second equality comes from that $\boldsymbol{y}$ and $\boldsymbol{x}_t$ are conditionally independent on $\boldsymbol{x}_0$, as shown in Fig. 3.2. Here, $p(\boldsymbol{x}_0|\boldsymbol{x}_t)$, as was shown with blue dotted lines in Fig. 3.2, is intractable in general. Note however, that for the case of diffusion

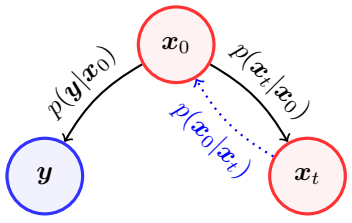


Figure 3.2: Probabilistic graph. Black solid line: tractable, blue dotted line: intractable in general.

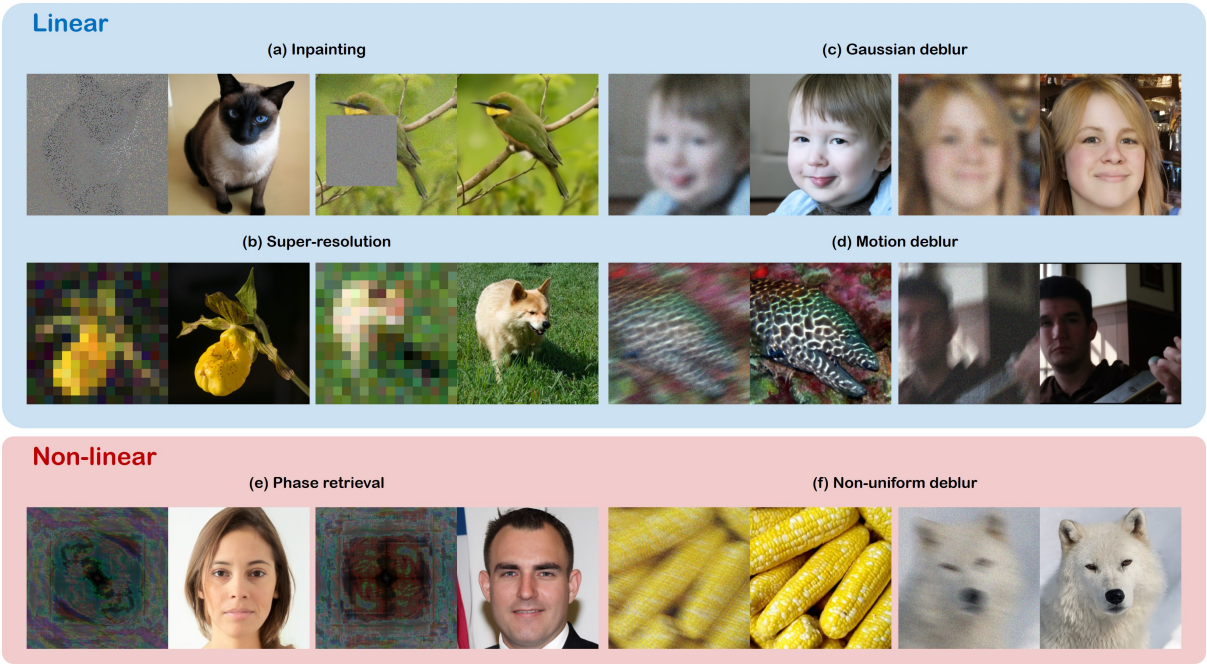


Figure 3.1: Solving noisy linear, and nonlinear inverse problems with diffusion models. Our reconstruction results with DPS (right) from the measurements (left) are shown.

models such as VP-SDE or DDPM, we have access to the posterior mean through Theorem 1

$$\hat{\boldsymbol{x}}_0 := \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t] = \frac{1}{\sqrt{\bar{\alpha}(t)}}(\boldsymbol{x}_t + (1-\bar{\alpha}(t))\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t)) \tag{3.4}$$

$$\approx \frac{1}{\sqrt{\bar{\alpha}(t)}}(\boldsymbol{x}_t + (1-\bar{\alpha}(t))\boldsymbol{s}_{\theta^*}(\boldsymbol{x}_t, t)), \tag{3.5}$$

where we have used a plug-in approximation with a pre-trained diffusion model $\boldsymbol{s}_{\theta^*}$.

Given the posterior mean $\hat{\boldsymbol{x}}_0$ that can be efficiently computed at the intermediate steps, our proposal is to provide a tractable approximation for $p(\boldsymbol{y}|\boldsymbol{x}_t)$ such that one can use the surrogate function to maximize the likelihood—yielding approximate posterior sampling. Specifically, given the interpretation $p(\boldsymbol{y}|\boldsymbol{x}_t) = \mathbb{E}_{\boldsymbol{x}_0\sim p(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[p(\boldsymbol{y}|\boldsymbol{x}_0)\right]$ from Eq. (3.3), we use the following approximation:

$$p(\boldsymbol{y}|\boldsymbol{x}_t) \simeq p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0), \quad \text{where} \quad \hat{\boldsymbol{x}}_0 := \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t] = \mathbb{E}_{\boldsymbol{x}_0\sim p(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[\boldsymbol{x}_0\right] \tag{3.6}$$

implying that the outer expectation of $p(\boldsymbol{y}|\boldsymbol{x}_0)$ over the posterior distribution is replaced with inner expectation of $\boldsymbol{x}_0$. In fact, this type of the approximation is closely related to the Jensen's inequality, so we need the following definition to quantify the approximation error:

Definition 1 (Jensen gap [53, 147])**.** *Let $\boldsymbol{x}$ be a random variable with distribution $p(\boldsymbol{x})$. For some function f that may or may not be convex, the Jensen gap is defined as*

$$\mathcal{J}(f, \boldsymbol{x}\sim p(\boldsymbol{x})) = \mathbb{E}[f(\boldsymbol{x})] - f(\mathbb{E}[\boldsymbol{x}]), \tag{3.7}$$

where the expectation is taken over $p(\boldsymbol{x})$.

The following theorem derives the closed-form upper bound of the Jensen gap for the inverse problem from Eq. (2.1):

Algorithm 1 DPS - Gaussian

Require: N, $\boldsymbol{y}$, $\{\zeta_i\}_{i=1}^N$, $\{\tilde{\sigma}_i\}_{i=1}^N$
1: $\boldsymbol{x}_N \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
2: **for** $i = N-1$ **to** 0 **do**
3: $\quad \hat{\boldsymbol{s}} \leftarrow \boldsymbol{s}_\theta(\boldsymbol{x}_i, i)$
4: $\quad \hat{\boldsymbol{x}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_i}}(\boldsymbol{x}_i + (1-\bar{\alpha}_i)\hat{\boldsymbol{s}})$
5: $\quad \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
6: $\quad \boldsymbol{x}'_{i-1} \leftarrow \frac{\sqrt{\alpha_i}(1-\bar{\alpha}_{i-1})}{1-\bar{\alpha}_i}\boldsymbol{x}_i + \frac{\sqrt{\bar{\alpha}_{i-1}}\beta_i}{1-\bar{\alpha}_i}\hat{\boldsymbol{x}}_0 + \tilde{\sigma}_i \boldsymbol{z}$
7: $\quad \boldsymbol{x}_{i-1} \leftarrow \boldsymbol{x}'_{i-1} - \zeta_i \nabla_{\boldsymbol{x}_i} \|\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{x}}_0)\|_2^2$
8: **end for**
9: **return** $\hat{\boldsymbol{x}}_0$

Algorithm 2 DPS - Poisson

Require: N, $\boldsymbol{y}$, $\{\zeta_i\}_{i=1}^N$, $\{\tilde{\sigma}_i\}_{i=1}^N$
1: $\boldsymbol{x}_N \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
2: **for** $i = N-1$ **to** 0 **do**
3: $\quad \hat{\boldsymbol{s}} \leftarrow \boldsymbol{s}_\theta(\boldsymbol{x}_i, i)$
4: $\quad \hat{\boldsymbol{x}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_i}}(\boldsymbol{x}_i + (1-\bar{\alpha}_i)\hat{\boldsymbol{s}})$
5: $\quad \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
6: $\quad \boldsymbol{x}'_{i-1} \leftarrow \frac{\sqrt{\alpha_i}(1-\bar{\alpha}_{i-1})}{1-\bar{\alpha}_i}\boldsymbol{x}_i + \frac{\sqrt{\bar{\alpha}_{i-1}}\beta_i}{1-\bar{\alpha}_i}\hat{\boldsymbol{x}}_0 + \tilde{\sigma}_i \boldsymbol{z}$
7: $\quad \boldsymbol{x}_{i-1} \leftarrow \boldsymbol{x}'_{i-1} - \zeta_i \nabla_{\boldsymbol{x}_i} \|\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{x}}_0)\|_{\boldsymbol{\Lambda}}^2$
8: **end for**
9: **return** $\hat{\boldsymbol{x}}_0$

Theorem 2. *For the given measurement model Eq. (2.1) with $\boldsymbol{n} \sim \mathcal{N}(0, \sigma_y^2 \boldsymbol{I})$, we have*

$$p(\boldsymbol{y}|\boldsymbol{x}_t) \simeq p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0), \tag{3.8}$$

where the approximation error can be quantified with the Jensen gap, which is upper bounded by

$$\mathcal{J} \leq \frac{d}{\sqrt{2\pi\sigma_y^2}} e^{-1/2\sigma_y^2} \|\nabla_{\boldsymbol{x}} \mathcal{A}(\boldsymbol{x})\| m_1, \tag{3.9}$$

where $\|\nabla_{\boldsymbol{x}} \mathcal{A}(\boldsymbol{x})\| := \max_{\boldsymbol{x}} \|\nabla_{\boldsymbol{x}} \mathcal{A}(\boldsymbol{x})\|$ and $m_1 := \int \|\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0\| p(\boldsymbol{x}_0|\boldsymbol{x}_t)\, d\boldsymbol{x}_0$.

Remark 1. *Note that $\|\nabla_{\boldsymbol{x}} \mathcal{A}(\boldsymbol{x})\|$ is finite in most of the inverse problems. This should not be confused with the ill-posedness of the inverse problems, which refers to the unboundedness of the inverse operator $\mathcal{A}^{-1}$. Accordingly, if m_1 is also finite (which is the case for most of the distribution in practice), the Jensen gap in Theorem 2 can approach 0 as $\sigma_y \to \infty$, suggesting that the approximation error reduces with higher measurement noise. This may explain why our DPS works well for noisy inverse problems. In addition, although we have specified the measurement distribution to be Gaussian, we can also determine the Jensen gap for other measurement distributions (e.g. Poisson) in an analogous fashion.*

By leveraging the result of Theorem 2, we can use the approximate gradient of the log likelihood

$$\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{y}|\boldsymbol{x}_t) \simeq \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0), \tag{3.10}$$

where the latter is now analytically tractable, as the measurement distribution is given.

3.2.2 Model dependent likelihood of the measurement

Note that we may have different measurement models $p(\boldsymbol{y}|\boldsymbol{x}_0)$ for each application. Two of the most common cases in inverse problems are the Gaussian noise and the Poisson noise. Here, we explore how our diffusion posterior sampling described above can be adapted to each case.

Gaussian noise. The likelihood function takes the form

$$p(\boldsymbol{y}|\boldsymbol{x}_0) = \frac{1}{\sqrt{(2\pi)^n \sigma_y^{2n}}} \exp\left[-\frac{\|\boldsymbol{y} - \mathcal{A}(\boldsymbol{x}_0)\|_2^2}{2\sigma_y^2}\right],$$

where n denotes the dimension of the measurement $\boldsymbol{y}$. By differentiating $p(\boldsymbol{y}|\boldsymbol{x}_t)$ with respect to $\boldsymbol{x}_t$, using

Theorem 2 and Eq. (3.10), we get

$$\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{y}|\boldsymbol{x}_t) \simeq -\frac{1}{2\sigma_y^2} \nabla_{\boldsymbol{x}_t} \|\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t))\|_2^2$$

where we have explicitly denoted $\hat{\boldsymbol{x}}_0 := \hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)$ to emphasize that $\hat{\boldsymbol{x}}_0$ is a function of $\boldsymbol{x}_t$. Consequently, taking the gradient $\nabla_{\boldsymbol{x}_t}$ amounts to taking the backpropagation through the network. Plugging in the result from Theorem 2 to Eq. (3.2) with the trained score function, we finally conclude that

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) \simeq \boldsymbol{s}_{\theta^*}(\boldsymbol{x}_t, t) - \rho \nabla_{\boldsymbol{x}_t} \|\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{x}}_0)\|_2^2, \tag{3.11}$$

where $\rho \triangleq 1/2\sigma^2$ is set as the step size. In practical implementation, we instead use ζ_i to express the step size. From the experiments, we observe that taking $\zeta_i = \zeta'/\|\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{x}}_0(\boldsymbol{x}_i))\|$, with ζ' set to constant, yields highly stable results.

Poisson noise. The likelihood function for the Poisson measurements under the i.i.d. assumption is given as

$$p(\boldsymbol{y}|\boldsymbol{x}_0) = \prod_{j=1}^{n} \frac{[\mathcal{A}(\boldsymbol{x}_0)]_j^{\boldsymbol{y}_j} \exp\left[[-\mathcal{A}(\boldsymbol{x}_0)]_j\right]}{\boldsymbol{y}_j\,!}, \tag{3.12}$$

where j indexes the measurement bin. In most cases where the measured values are not too small, the model can be approximated by a Gaussian distribution with very high accuracy. Namely,

$$p(\boldsymbol{y}|\boldsymbol{x}_0) \to \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi[\mathcal{A}(\boldsymbol{x}_0)]_j}} \exp\left(-\frac{(\boldsymbol{y}_j - [\mathcal{A}(\boldsymbol{x}_0)]_j)^2}{2[\mathcal{A}(\boldsymbol{x}_0)]_j}\right) \tag{3.13}$$

$$\simeq \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi \boldsymbol{y}_j}} \exp\left(-\frac{(\boldsymbol{y}_j - [\mathcal{A}(\boldsymbol{x}_0)]_j)^2}{2\boldsymbol{y}_j}\right), \tag{3.14}$$

where we have used the standard approximation for the shot noise model $[\mathcal{A}(\boldsymbol{x}_0)]_j \simeq \boldsymbol{y}_j$ to arrive at the last equation [90]. Then, similar to the Gaussian case, by differentiation and the use of Theorem 2, we have that

$$\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{y}|\boldsymbol{x}_t) \simeq -\rho \nabla_{\boldsymbol{x}_t} \|\boldsymbol{y} - \mathcal{A}(\boldsymbol{x}_0)\|_{\boldsymbol{\Lambda}}^2, \quad [\boldsymbol{\Lambda}]_{ii} \triangleq 1/2\boldsymbol{y}_j, \tag{3.15}$$

where $\|\boldsymbol{a}\|_{\boldsymbol{\Lambda}}^2 \triangleq \boldsymbol{a}^T \boldsymbol{\Lambda} \boldsymbol{a}$, and we have included ρ to define the step size as in the Gaussian case. We can summarize our strategy for each noise model as follows:

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) \simeq s_{\theta^*}(\boldsymbol{x}_t, t) - \rho \nabla_{\boldsymbol{x}_t} \|\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{x}}_0)\|_2^2 \qquad \text{(Gaussian)} \tag{3.16}$$

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) \simeq s_{\theta^*}(\boldsymbol{x}_t, t) - \rho \nabla_{\boldsymbol{x}_t} \|\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{x}}_0)\|_{\boldsymbol{\Lambda}}^2 \qquad \text{(Poisson)} \tag{3.17}$$

Incorporation of Eq. (3.11) or Eq. (3.16) into the usual ancestral sampling [62] steps leads to Algorithm 1,2 Here, we name our algorithm **D**iffusion **P**osterior **S**ampling (DPS), as we construct our method in order to perform sampling from the posterior distribution. Notice that, unlike prior methods that limit their applications to linear inverse problems $\mathcal{A}(\boldsymbol{x}) \triangleq \boldsymbol{A}\boldsymbol{x}$, our method is fully general in that we can also use nonlinear operators $\mathcal{A}(\cdot)$. To show that this is indeed the case, in the experimental section we take the two notoriously hard nonlinear inverse problems: Fourier phase retrieval and non-uniform deblurring, and show that our method has very strong performance even in such challenging problem settings.

Method	SR (×4)		Inpaint (box)		Inpaint (random)		Deblur (gauss)		Deblur (motion)	
	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓
DPS (ours)	**39.35**	**0.214**	**33.12**	**0.168**	**21.19**	**0.212**	**44.05**	**0.257**	**39.92**	**0.242**
DDRM [82]	62.15	0.294	42.93	0.204	69.71	0.587	74.92	0.332	-	-
MCG [29]	87.64	0.520	40.11	0.309	29.26	0.286	101.2	0.340	310.5	0.702
PnP-ADMM [17]	66.52	0.353	151.9	0.406	123.6	0.692	90.42	0.441	89.08	0.405
Score-SDE [156] (ILVR [23])	96.72	0.563	60.06	0.331	76.54	0.612	109.0	0.403	292.2	0.657
ADMM-TV	110.6	0.428	68.94	0.322	181.5	0.463	186.7	0.507	152.3	0.508

Table 3.1: Quantitative evaluation (FID, LPIPS) of solving linear inverse problems on FFHQ 256×256-1k validation dataset. **Bold**: best, underline: second best.

Method	SR (×4)		Inpaint (box)		Inpaint (random)		Deblur (gauss)		Deblur (motion)	
	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓
DPS (ours)	**50.66**	**0.337**	**38.82**	0.262	**35.87**	**0.303**	**62.72**	0.444	**56.08**	**0.389**
DDRM [82]	59.57	0.339	45.95	**0.245**	114.9	0.665	63.02	**0.427**	-	-
MCG [29]	144.5	0.637	39.74	0.330	39.19	0.414	95.04	0.550	186.9	0.758
PnP-ADMM [17]	97.27	0.433	78.24	0.367	114.7	0.677	100.6	0.519	89.76	0.483
Score-SDE [156] (ILVR [23])	170.7	0.701	54.07	0.354	127.1	0.659	120.3	0.667	98.25	0.591
ADMM-TV	130.9	0.523	87.69	0.319	189.3	0.510	155.7	0.588	138.8	0.525

Table 3.3: Quantitative evaluation (FID, LPIPS) of solving linear inverse problems on ImageNet 256×256-1k validation dataset. **Bold**: best, underline: second best.

3.2.3 Experiments

Experimental setup. We test our experiment on two datasets that have diverging characteristic - FFHQ 256×256 [80], and Imagenet 256×256 [37], on 1k validation images each. The pre-trained diffusion model for ImageNet was taken from [39] and was used directly without finetuning for specific tasks. The diffusion model for FFHQ was trained from scratch using 49k training data (to exclude 1k validation set) for 1M steps. All images are normalized to the range $[0, 1]$. Forward measurement operators are specified as follows: (i) For box-type inpainting, we mask out 128×128 box region following [29], and for random-type we mask out 92% of the total pixels (all RGB channels). (ii) For super-resolution, bicubic downsampling is performed. (iii) Gaussian blur kernel has size 61×61 with standard deviation of 3.0, and motion blur is randomly generated with the code[1], with size 61×61 and intensity value 0.5. The kernels are convolved with the ground truth image to produce the measurement. (iv) For phase retrieval, Fourier transform is performed to the image, and only the Fourier magnitude is taken as the measurement. (v) For nonlinear deblurring, we leverage the neural network approximated forward model as in [160]. All Gaussian noise is added to the measurement domain with $\sigma = 0.05$. Poisson noise level is set to $\lambda = 1.0$.

Method	FID ↓	LPIPS ↓
DPS(ours)	**55.61**	**0.399**
OSS	137.7	0.635
HIO	96.40	0.542
ER	214.1	0.738

Table 3.2: Quantitative evaluation of the Phase Retrieval task (FFHQ).

We perform comparison with the following methods: Denoising diffusion restoration models (DDRM) [82], manifold constrained gradients (MCG) [29], Plug-and-play alternating direction method of multipliers (PnP-

[1] https://github.com/LeviBorodenko/motionblur

Figure 3.3: Results on solving linear inverse problems with Gaussian noise ($\sigma = 0.05$).

ADMM) [17] using DnCNN [179] in place of proximal mappings, total-variation (TV) sparsity regularized optimization method (ADMM-TV), and Score-SDE [156]. Note that [156] only proposes a method for inpainting, and not for general inverse problems. However, the methodology of iteratively applying projections onto convex sets (POCS) was applied in the same way for super-resolution in iterative latent variable refinement (ILVR) [23], and more generally to linear inverse problems in [30]; thus we simply refer to these methods as score-SDE henceforth.For a fair comparison, we used the same score function for all the different methods that are based on diffusion (i.e. DPS, DDRM, MCG, score-SDE).

For phase retrieval, we compare with three strong baselines that are considered standards: oversampling smoothness (OSS) [134], Hybrid input-output (HIO) [48], and error reduction (ER) algorithm [50]. For nonlinear deblurring, we compare against the prior arts: blur kernel space (BKS) - styleGAN2 [160], based on GAN priors, blur kernel space (BKS) - generic [160], based on Hyper-Laplacian priors, and MCG. For quantitative comparison, we focus on the following two widely-used perceptual metrics - Fréchet Inception Distance (FID), and Learned Perceptual Image Patch Similarity (LPIPS) distance.

Noisy linear inverse problems. We first test our method on diverse linear inverse problems with Gaussian measurement noises. The quantitative results shown in Tables 3.1,3.3 illustrate that the proposed method outperforms all the other comparison methods by large margins. Particularly, MCG and Score-SDE (or ILVR) are methods that rely on projections on the measurement subspace, where the generative process is controlled such that the measurement consistency is *perfectly* met. While this is useful for noiseless (or negligible noise) problems, in the case where we cannot ignore noise, the solutions overfit to the corrupted measurement. In Fig. 3.3, we specifically compare our methods with DDRM and PnP-ADMM, which are two methods that are known to be robust to measurement noise. Our method is able to provide high-quality reconstructions that are crisp and realistic on all tasks. On the other hand,

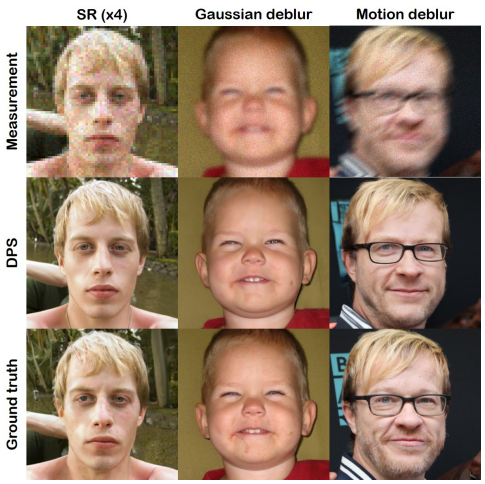


Figure 3.4: Results on solving linear inverse problems with Poisson noise ($\lambda = 1.0$).

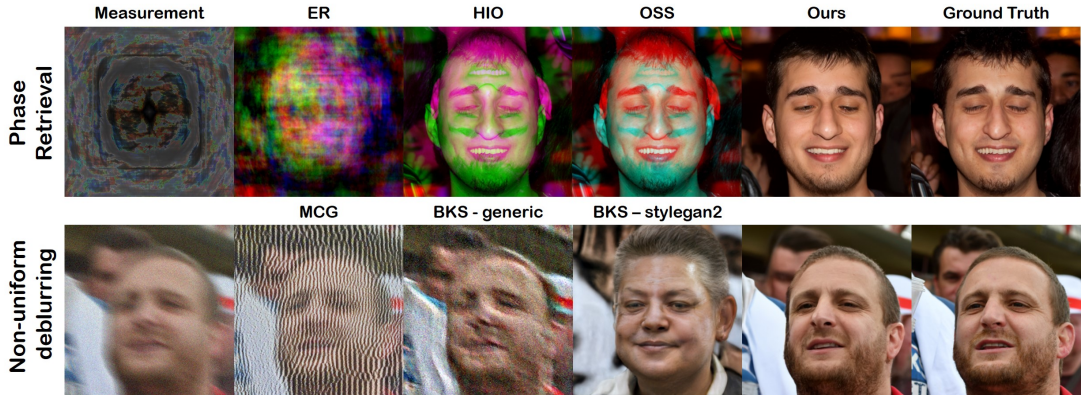


Figure 3.5: Results on solving nonlinear inverse problems with Gaussian noise ($\sigma = 0.05$).

we see that DDRM performs poorly on image inpainting tasks where the dimensionality of the measurements are very low, and tend to produce blurrier results on both SR, and deblurring tasks. We further note that DDRM relies on SVD, and hence is only able to solve problems where the forward measurement matrix can be efficiently implemented (e.g. separable kernel in the case of deblurring). Hence, while one can solve Gaussian deblurring, one cannot solve problems such as motion deblur, where the point spread function (PSF) is much more complex. Contrarily, our method is not restricted by such conditions, and can be always used regardless of the complexity.

The results of the Poisson noisy linear inverse problems are presented in Fig. 3.4. Consistent with the Gaussian case, DPS is capable of producing high quality reconstructions that closely mimic the ground truth. From the experiments, we further observe that the weighted least squares method adopted in Algorithm 2 works best compared to other choices that can be made for Poisson inverse problems.

Nonlinear inverse problems. We show the quantitative results of phase retrieval in Table 3.2, and the results of nonlinear deblurring in Table 3.4. Representative results are illustrated in Fig. 3.5. We first observe that the proposed method is capable of highly accurate reconstruction for the given phase retrieval problem, capturing most of the high frequency details. However, we also observe that we do not *always* get high quality reconstructions. In fact, due to the non-uniqueness of the phase-retrieval under some conditions, widely used methods such as HIO are also dependent on the initializations [49], and hence it is considered standard practice to first generate multiple reconstructions, and take the best sample. Following this, when reporting our quantitative metrics, we generate 4 different samples for all the methods, and report the metric based on the best samples. We see that DPS outperforms other methods by a large margin. For the case of nonlinear deblurring, we again see that our method performs the best, producing highly realistic samples. BKS-styleGAN2 [160] leverages GAN prior and hence generates feasible human faces, but heavily distorts the identity. BKS-generic utilizes the Hyper-Laplacian prior [94], but is unable to remove artifacts and noise properly.

Method	FID ↓	LPIPS ↓
DPS(ours)	**41.86**	0.278
BKS-styleGAN2	63.18	0.407
BKS-generic	141.0	0.640
MCG	180.1	0.695

Table 3.4: Quantitative evaluation of the non-uniform deblurring task (FFHQ).

3.3 BlindDPS: Blind extension of DPS

Notice that in Sec. 3.2, we only considered the non-blind case within process information. The blind case considers the cases where the operator is *unknown*, and thus the operator needs to be estimated together with the reconstruction of the latent image. The latter problem is considerably harder than the former problem, as joint

Figure 3.6: Representative results and overall concept of BlindDPS. (a) Results of blind deblurring. Both the image and the kernel in the bottom right corner are jointly estimated with the proposed method. (b) Results of imaging through turbulence. (c) Evolution of joint reconstruction with the proposed method. 1^{st}, 2^{nd} row illustrate the change of $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)$ and $\hat{\boldsymbol{k}}_0(\boldsymbol{k}_t)$ through time as $t = 1 \to 0$, with the measurement and the kernel initialization given on the first column.

minimization is typically much less stable.

Consider blind deblurring, a canonical blind inverse problem studied in computer vision. In such cases, not only do we need a prior model of the image, but we also need some proper prior model of the kernel [120, 159]. While conventional methods exploit, e.g. patch-based prior [159], sparsity prior [120], etc., they often fall short of accurate modeling of the distribution. Here, we aim to leverage the ability of diffusion models to act as strong generative priors and propose *BlindDPS* (Blind Diffusion Posterior Sampling) — constructing multiple diffusion processes for learning the prior of each component — which enable posterior sampling even when the operator is unknown. BlindDPS starts by initializing both the image and the operator parameter with Gaussian noise. Reverse diffusion progresses in parallel for both models, where the cross-talk between the paths are enforced from the approximate likelihood and the measurement, as can be seen in Fig. 3.7. With our method, both the image and the kernel starts with a coarse estimation, gradually getting closer to the ground truth as $t \to 0$ (see Fig. 3.6(c)).

In fact, our method can be thought of as a coarse-to-fine strategy naturally admitting a Gaussian scale-space representation [92, 106], which can be seen as a continuous generalization of the coarse-to-fine optimization strategy that most of the optimization-based methods take [120, 122]. Furthermore, our method is generally applicable to cases where we know the *structure* of the forward model a priori (e.g. convolution). To demonstrate the generality, we further show that our method can also be applied in imaging through turbulence. From our experiments, we show that the proposed method yields state-of-the-art performance while being generalizable to different inverse problems.

Blind inverse problems Blind inverse problems consider the case where the forward model $\mathcal{A}$ is unknown. Among them, we focus on the case where the forward operator is parameterized with $\boldsymbol{\varphi}$, and we need to estimate

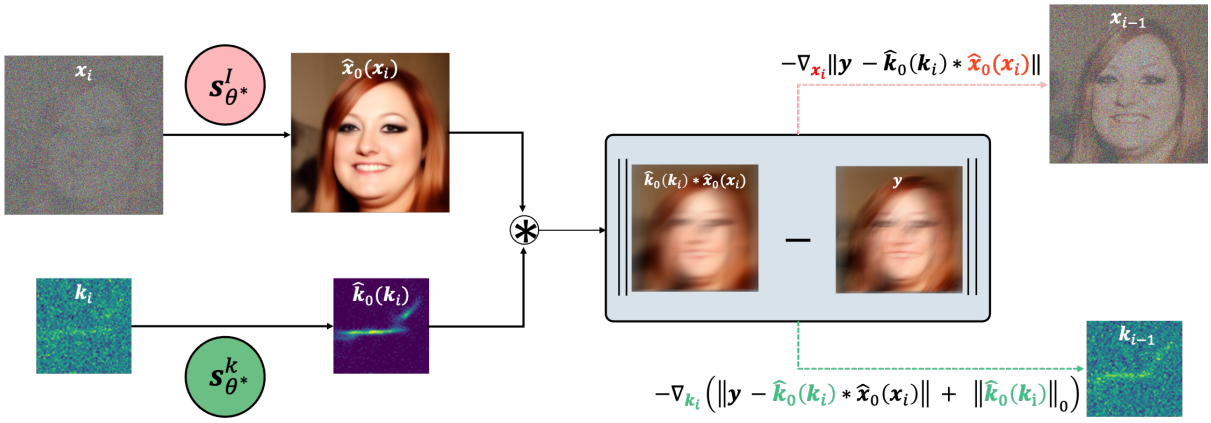


Figure 3.7: Description of BlindDPS. From the intermediate (noisy) estimate $\boldsymbol{x}_i, \boldsymbol{k}_i$, we achieve the denoised representation $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_i), \hat{\boldsymbol{k}}_0(\boldsymbol{k}_i)$ through Tweedie's formula with the score functions $\boldsymbol{s}^i_{\theta^*}, \boldsymbol{s}^k_{\theta^*}$. The residual $\|\boldsymbol{y} - \hat{\boldsymbol{k}}_0 * \hat{\boldsymbol{x}}_0\|$ is computed with the denoised estimates, and the residual-minimizing gradients are applied parallel to both diffusion processes.

the *parameter* $\boldsymbol{\varphi}$. Specifically, consider the following forward model

$$\boldsymbol{y} = \mathcal{A}_{\boldsymbol{\varphi}}(\boldsymbol{x}) + \boldsymbol{n}, \tag{3.18}$$

where $\boldsymbol{\varphi}$ is the parameter of the forward model, $\boldsymbol{x}$ is the ground truth image, and $\boldsymbol{n}$ is some noise. Here, both $\boldsymbol{\varphi}, \boldsymbol{x}$ are unknown, and should be estimated. A classical way to solve Eq. (3.18) is to optimize for the following

$$\min_{\boldsymbol{x},\boldsymbol{\varphi}} \quad \frac{1}{2}\|\mathcal{A}_{\boldsymbol{\varphi}}(\boldsymbol{x}) - \boldsymbol{y}\|^2 + R_{\boldsymbol{\varphi}}(\boldsymbol{\varphi}) + R_{\boldsymbol{x}}(\boldsymbol{x}), \tag{3.19}$$

where $R_{\boldsymbol{\varphi}}(\boldsymbol{\varphi}), R_{\boldsymbol{x}}(\boldsymbol{x})$ are regularization functions for $\boldsymbol{\varphi}, \boldsymbol{x}$, respectively, which can also be thought of as the negative log prior for each distribution, e.g. $R(\cdot) = -\log p(\cdot)$.

For example, consider blind deconvolution from camera motion blur as illustrated in Fig. 3.8(a). The forward model reads

$$\boldsymbol{y} = \boldsymbol{k} * \boldsymbol{x} + \boldsymbol{n}, \tag{3.20}$$

where $\boldsymbol{k}$ is the blur kernel, corresponding to the parameter $\boldsymbol{\varphi}$. On the other hand, although the "real" forward model for atmospheric turbulence is rarely directly used in practice due to the highly complicated nature of the wave propagation theory, the tilt-blur model is often used [16, 144, 15], as the model is simple but fairly accurate. Specifically, the visualization of such imaging process is shown in Fig. 3.8(b), which can be mathematically described by

$$\boldsymbol{y} = \boldsymbol{k} * \mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{x}) + \boldsymbol{n}, \tag{3.21}$$

where $\mathcal{T}$ is the tilt operator parameterized by the tilt vector field $\boldsymbol{\phi}$. To remove the scale ambiguity between the kernel and image, the magnitude and the polarity constraints of kernels are often used:

$$\mathbf{1}^T\boldsymbol{k} = 1, \boldsymbol{k} \succeq 0. \tag{3.22}$$

Then, the success of the optimization algorithm Eq. (3.19) with the forward models Eq. (3.20) or Eq. (3.21) under the constraint Eq. (3.22) depends on two factors: 1) How closely the prior-imposing functions $R_{\{\boldsymbol{x},\boldsymbol{k}\}}$ estimate

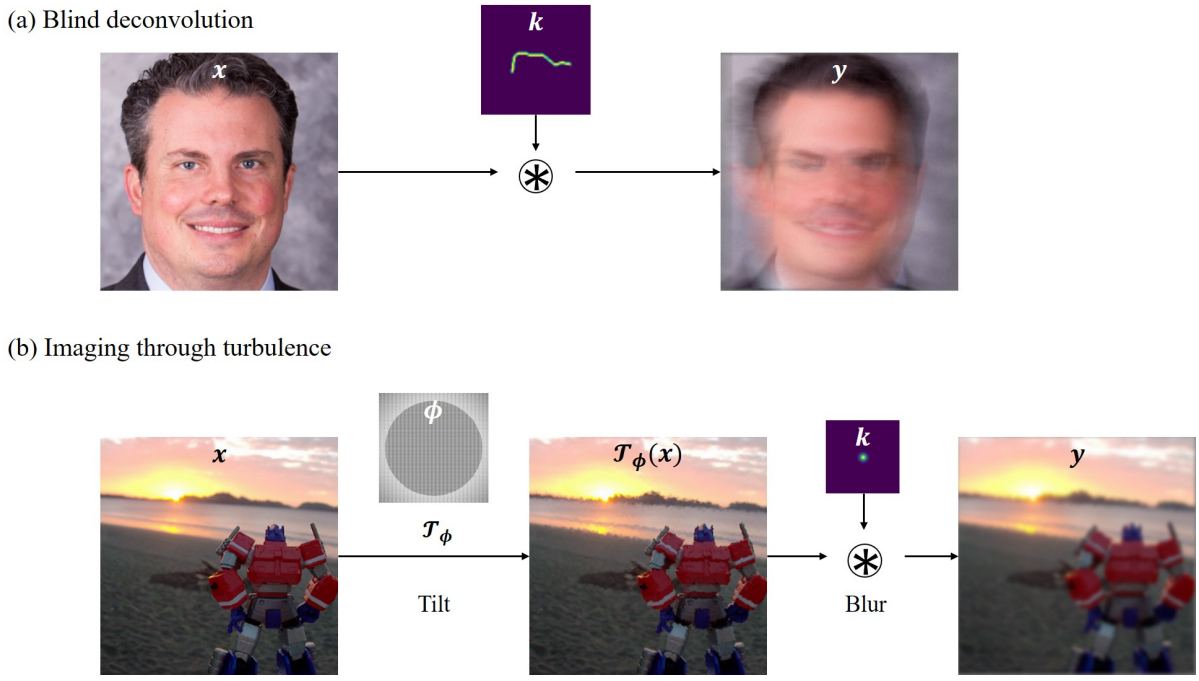


Figure 3.8: Illustration of the imaging forward model. (a) Blind deconvolution, (b) Imaging through turbulence

the true prior, and 2) how well the optimization procedure finds the minimum value. Conventional methods are sub-optimal in both aspects. First, the prior (e.g. sparsity [120], dark channel [122], implicit from deep networks [132]) functions do not fully represent the *true* prior. Second, the optimization process is unstable and hard to tune. For instance, [120, 122] requires different weighting parameters *per image*, and often fails during the abrupt changes in the stage transition during coarse-to-fine optimization strategy.

Key idea In DPS, we used the diffusion prior for $R_{\boldsymbol{x}}$ by training a score function that models $\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})$. As for blind inverse problems, a prior model for the parameter $p(\boldsymbol{\varphi})$ should also be specified. In this regard, our proposal is to use the diffusion prior also for the forward model parameter by estimating $\nabla_{\boldsymbol{\varphi}} \log p(\boldsymbol{\varphi})$. With such choice, one can model a much more accurate prior for the parameters compared to the conventional choices. In the following, we detail on how to build our method *BlindDPS*, focusing on blind deconvolution. The method for imaging through turbulence can be derived in a completely analogous fashion.

In blind deblurring (deconvolution), the probabilistic forward model is specified as follows

$$p(\boldsymbol{y}|\boldsymbol{x}_0, \boldsymbol{k}_0) := \mathcal{N}(\boldsymbol{y}|\boldsymbol{k}_0 * \boldsymbol{x}_0, \sigma^2 \boldsymbol{I}), \tag{3.23}$$

where $\boldsymbol{k}_0$ is the random variable of the convolution kernel. As $\boldsymbol{x}_0$ and $\boldsymbol{k}_0$ are independent, the posterior probability is given as

$$p(\boldsymbol{x}_0, \boldsymbol{k}_0|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{x}_0, \boldsymbol{k}_0) p(\boldsymbol{x}_0) p(\boldsymbol{k}_0). \tag{3.24}$$

Note that our aim is to use implicit diffusion priors for both $p(\boldsymbol{x}_0)$ and $p(\boldsymbol{k}_0)$ through their score functions. One can easily take pre-trained score functions for the image. Similarly, the score function for the kernel can also be estimated from standard DSM Eq. (2.17) to get $\boldsymbol{s}^k_{\theta^*}(\boldsymbol{k}, t) \simeq \nabla_{\boldsymbol{k}_t} \log p_t(\boldsymbol{k}_t)$. Note that performing DSM to achieve $\boldsymbol{s}^k_{\theta^*}$ costs much less than training the image score function $\boldsymbol{s}^i_{\theta^*}$, as the distribution is much simpler, and the dimensionality of the vector $\boldsymbol{k}$ is also sufficiently smaller than $\boldsymbol{x}$.

On the other hand, again from the independence of $\boldsymbol{x}_0$ and $\boldsymbol{k}_0$, we are able to construct two separate reverse diffusion processes of identical form:

$$d\boldsymbol{x} = \left[-\frac{\beta(t)}{2}\boldsymbol{x} - \beta(t)\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t)\right]dt + \sqrt{\beta(t)}d\bar{\boldsymbol{w}}, \tag{3.25}$$

$$d\boldsymbol{k} = \left[-\frac{\beta(t)}{2}\boldsymbol{k} - \beta(t)\nabla_{\boldsymbol{k}_t}\log p(\boldsymbol{k}_t)\right]dt + \sqrt{\beta(t)}d\bar{\boldsymbol{w}}. \tag{3.26}$$

Note that the two reverse SDEs are only able to sample from the marginals — $p(\boldsymbol{x}_0)$, $p(\boldsymbol{k}_0)$. However, one can define the dependency between $\boldsymbol{x}, \boldsymbol{y}$, and $\boldsymbol{k}$ from the posterior probability. Using Bayes' rule in Eq. (3.24) for general t, we have

$$\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t, \boldsymbol{k}_t|\boldsymbol{y}) = \nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{y}|\boldsymbol{x}_t, \boldsymbol{k}_t) + \nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t), \tag{3.27}$$

$$\nabla_{\boldsymbol{k}_t}\log p(\boldsymbol{x}_t, \boldsymbol{k}_t|\boldsymbol{y}) = \nabla_{\boldsymbol{k}_t}\log p(\boldsymbol{y}|\boldsymbol{x}_t, \boldsymbol{k}_t) + \nabla_{\boldsymbol{k}_t}\log p(\boldsymbol{k}_t). \tag{3.28}$$

Here, in order to estimate the time-conditional log-likelihood $\log p(\boldsymbol{y}|\boldsymbol{x}_t, \boldsymbol{k}_t)$ which is intractable in general, we need the following result:

Theorem 3 (informal)**.** *With similar approximation error as in Theorem 2,*

$$\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{y}|\boldsymbol{x}_t, \boldsymbol{k}_t) \simeq \nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t), \hat{\boldsymbol{k}}_0(\boldsymbol{k}_t))$$
$$\nabla_{\boldsymbol{k}_t}\log p_t(\boldsymbol{y}|\boldsymbol{x}_t, \boldsymbol{k}_t) \simeq \nabla_{\boldsymbol{k}_t}\log p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t), \hat{\boldsymbol{k}}_0(\boldsymbol{k}_t)).$$

Remark 2. *Our theorem holds as long as $\boldsymbol{x}_t$, $\boldsymbol{k}_t$ are independent. Note that the theorem can be further generalized to handle more random variables whenever the independence between the variables is established. In other words, we can construct arbitrary many diffusion procedures for each component of the forward model, which can be solved analogous to the approximation proposed in Theorem 3. This result will be useful, for instance, when we solve the problem of imaging through turbulence.*

Using Theorem 3, we finally arrive at the following reverse SDEs

$$\begin{aligned} d\boldsymbol{x} = (&-\frac{\beta(t)}{2}\boldsymbol{x} - \beta(t)[\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t), \hat{\boldsymbol{k}}_0(\boldsymbol{k}_t)) \\ &+ \boldsymbol{s}^i_{\theta^*}(\boldsymbol{x}_t, t)])dt + \sqrt{\beta(t)}d\bar{\boldsymbol{w}}, \end{aligned} \tag{3.29}$$

$$\begin{aligned} d\boldsymbol{k} = (&-\frac{\beta(t)}{2}\boldsymbol{k} - \beta(t)[\nabla_{\boldsymbol{k}_t}\log p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t), \hat{\boldsymbol{k}}_0(\boldsymbol{k}_t)) \\ &+ \boldsymbol{s}^k_{\theta^*}(\boldsymbol{k}_t, t)])dt + \sqrt{\beta(t)}d\bar{\boldsymbol{w}}. \end{aligned} \tag{3.30}$$

The system of equations Eq. (3.29),Eq. (3.30) are now numerically solvable as the gradient of the log likelihood is analytically tractable. Specifically, for the Gaussian measurement, we have

$$\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0, \hat{\boldsymbol{k}}_0) = -\frac{1}{\sigma^2}\nabla_{\boldsymbol{x}_t}\|\boldsymbol{y} - \hat{\boldsymbol{k}}_0 * \hat{\boldsymbol{x}}_0\|_2^2. \tag{3.31}$$

Combined with the ancestral sampling steps [62], our algorithm for posterior sampling of blind deblurring is formally given in Algorithm 3. Here, note that we choose to take static step size times the gradient of the norm instead of taking time-dependent step sizes times the gradient of the squared norm, as it was shown to be effective despite its simplicity [26]. Furthermore, in order to impose the usual condition Eq. (3.22), we define a set $C := \{\boldsymbol{k}|\mathbf{1}^T\boldsymbol{k} = 1, \boldsymbol{k} \succeq 0\}$, and project onto the set through $\mathcal{P}_C(\hat{\boldsymbol{k}}_0)$ in Algorithm 3, after the estimation of $\hat{\boldsymbol{k}}_0$ at

Algorithm 3 BlindDPS — Blind Deblurring

Require: N, $\boldsymbol{y}$, α, $\{\tilde{\sigma}_i\}_{i=1}^N$, λ, $R_{\boldsymbol{k}}(\cdot)$
$\boldsymbol{x}_N, \boldsymbol{k}_N \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $i = N-1$ **to** 0 **do**
  $\hat{\boldsymbol{s}}^i \leftarrow \boldsymbol{s}^i_{\theta^*}(\boldsymbol{x}_i, i)$
  $\hat{\boldsymbol{s}}^k \leftarrow \boldsymbol{s}^k_{\theta^*}(\boldsymbol{k}_i, i)$
  $\hat{\boldsymbol{x}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_i}}(\boldsymbol{x}_i + \sqrt{1-\bar{\alpha}_i}\hat{\boldsymbol{s}}^i)$
  $\hat{\boldsymbol{k}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_i}}(\boldsymbol{k}_i + \sqrt{1-\bar{\alpha}_i}\hat{\boldsymbol{s}}^k)$
  $\hat{\boldsymbol{k}}_0 \leftarrow \mathcal{P}_C(\hat{\boldsymbol{k}}_0)$
  $\boldsymbol{z}_i, \boldsymbol{z}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\boldsymbol{x}'_{i-1} \leftarrow \frac{\sqrt{\alpha_i}(1-\bar{\alpha}_{i-1})}{1-\bar{\alpha}_i}\boldsymbol{x}_i + \frac{\sqrt{\bar{\alpha}_{i-1}}\beta_i}{1-\bar{\alpha}_i}\hat{\boldsymbol{x}}_0 + \tilde{\sigma}_i \boldsymbol{z}_i$
  $\boldsymbol{k}'_{i-1} \leftarrow \frac{\sqrt{\alpha_i}(1-\bar{\alpha}_{i-1})}{1-\bar{\alpha}_i}\boldsymbol{k}_i + \frac{\sqrt{\bar{\alpha}_{i-1}}\beta_i}{1-\bar{\alpha}_i}\hat{\boldsymbol{k}}_0 + \tilde{\sigma}_i \boldsymbol{z}_k$
  $\boldsymbol{x}_{i-1} \leftarrow \boldsymbol{x}'_{i-1} - \alpha \nabla_{\boldsymbol{x}_i} \|\boldsymbol{y} - \hat{\boldsymbol{k}}_0 * \hat{\boldsymbol{x}}_0\|_2$
  $\mathcal{L}_{\boldsymbol{k}} \leftarrow \|\boldsymbol{y} - \hat{\boldsymbol{k}}_0 * \hat{\boldsymbol{x}}_0\|_2 + \lambda R_{\boldsymbol{k}}(\hat{\boldsymbol{k}}_0)$
  $\boldsymbol{k}_{i-1} \leftarrow \boldsymbol{k}'_{i-1} - \alpha \nabla_{\boldsymbol{k}_i} \mathcal{L}_{\boldsymbol{k}}$
end for
return $\boldsymbol{x}_0, \boldsymbol{k}_0$

each intermediate step. For visual illustration of the proposed method, see Fig. 3.7.

Augmenting diffusion prior with sparsity Implementing Eq. (3.29),Eq. (3.30) directly induces fairly stable results with the correct choice of α. Here, we go a step further and adopt a lesson from the classic literature. As we often wish to estimate blur kernels that are sparse, we promote sparsity *only* to the kernel that we are estimating by augmenting the diffusion prior with ℓ_0/ℓ_1 regularization. The minimization strategy for the kernel then becomes

$$\boldsymbol{k}_{i-1} = \boldsymbol{k}'_{i-1} - \alpha \left(\|\boldsymbol{y} - \hat{\boldsymbol{k}}_0 * \hat{\boldsymbol{x}}_0\|_2 + \lambda R_{\boldsymbol{k}}(\hat{\boldsymbol{k}}_0) \right), \tag{3.32}$$

where λ is the regularization strength, and the choice of $R_{\boldsymbol{k}}(\cdot) := \ell_0/\ell_1$ regularization depends on the type of the dataset. With such augmentation, reconstruction can be further stabilized.

Interpretation in Gaussian scale-space (Gaussian) Scale-space theory [106] states that one can represent signals in multiple scales by gradually convolving with Gaussian filters. As adding Gaussian noise to random vectors in the forward pass of the diffusion has a dual relation in the density domain (i.e. convolution with Gaussian kernels), one can think of the diffusion process as a realization of one such process. Thus, the reverse diffusion process can be interpreted as a coarse-to-fine synthesis evolving through the Gaussian scale-space, which is most visible by visualizing $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t), \hat{\boldsymbol{k}}_0(\boldsymbol{k}_t)$ when evolving through $t = 1 \to 0$ (see Fig. 3.6(c)).

For blind deconvolution problems, in order to achieve optimal quality, it is a standard practice to start the optimization process at a coarse scale by down-sampling, and sequentially upsample with a pre-determined schedule to refine the estimates [121, 120]. However, the discretized schedule is typically abrupt (e.g. [121, 120] uses 8 discretization) and ad-hoc. On the other hand, by using the reverse diffusion process, we are granted with a natural, smooth schedule of evolution, which can be thought of as a continuous generalization of the coarse-to-fine reconstruction strategy. This could be another reason why the proposed method is able to dramatically outperform the conventional methods.

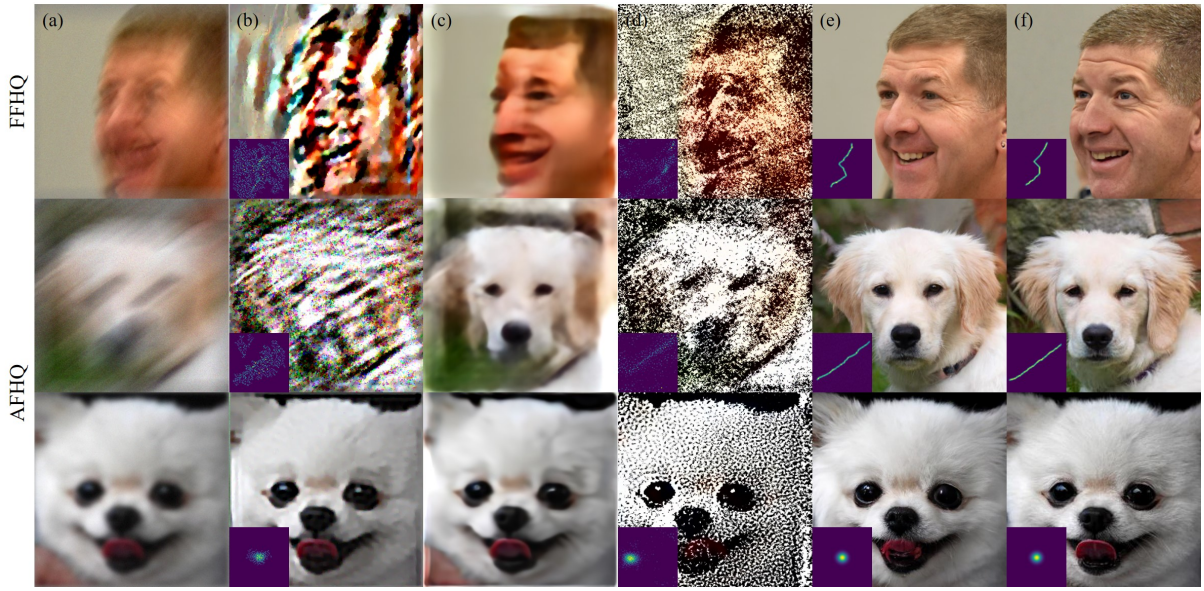


Figure 3.9: Blind deblurring results. (row 1): FFHQ 256×256 motion deblurring, (row 2): AFHQ 256×256 motion deblurring. (row 3): AFHQ 256×256 Gaussian deblurring. (a) Measurement, (b) Pan-DCP [122], (c) MPRNet [176], (d) SelfDeblur [132], (e) BlindDPS (**ours**), (f) Ground truth. For (c), kernel not shown as the method only estimate images.

	FFHQ (256 × 256)						**AFHQ (256 × 256)**					
	Motion			**Gaussian**			**Motion**			**Gaussian**		
Method	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑
BlindDPS (ours)	**29.49**	**0.281**	**22.24**	**27.36**	**0.233**	**24.77**	**23.89**	**0.338**	**20.92**	**20.54**	**0.287**	**23.63**
SelfDeblur [132]	270.0	0.717	10.83	235.4	0.686	11.36	300.5	0.768	9.081	172.2	0.662	11.53
MPRNet [176]	111.6	0.434	17.40	95.12	0.337	20.75	131.8	0.521	14.85	46.19	0.366	20.51
DeblurGANv2 [95]	220.7	0.571	17.75	185.5	0.529	19.69	186.2	0.597	17.64	86.87	0.523	20.29
Pan-DCP [122]	214.9	0.520	15.41	92.70	0.393	20.50	214.0	0.704	11.87	57.14	0.392	20.97
Pan-ℓ_0 [120]	242.6	0.542	15.53	109.1	0.415	19.94	235.0	0.627	15.34	62.76	0.395	21.41
Perrone *et al.* [126]	156.8	0.492	16.08	85.3	0.363	20.66	197.7	0.588	16.68	59.32	0.375	21.38

Table 3.5: Quantitative evaluation (FID, LPIPS, PSNR) of blind deblurring task on FFHQ and AFHQ. **Bold**: Best, under: second best.

3.3.1 Experiments

Dataset For blind deblurring, we use FFHQ 256×256 [80], and AFHQ-dog 256×256 [24]. We choose 1k validation set for FFHQ, and the full 500 test images for AFHQ-dog. We leverage pre-trained (image) score functions, as in the experimental setting of [30]. For imaging through turbulence, we use FFHQ 256×256 and ImageNet 256×256 [37]. For both blind inverse problems, we add Gaussian measurement noise with $\sigma = 0.02$.

Evaluation We use three metrics—Frechet inception distance (FID), learned Perceptual Image Patch Similarity (LPIPS), and peak signal-to-noise-ratio (PSNR)—for quantitatively measuring the performance of the image reconstruction. For kernel estimation, we use mean-squared-error (MSE), and maximum of normalized convolution (MNC) [68] , which is computed by

$$\text{MNC} := \max\left(\frac{\tilde{\boldsymbol{k}} * \boldsymbol{k}^*}{\|\tilde{\boldsymbol{k}}\|_2 \|\boldsymbol{k}^*\|_2}\right), \tag{3.33}$$

	FFHQ (256 × 256)				AFHQ (256 × 256)			
	Motion		Gaussian		Motion		Gaussian	
Method	MSE ↓	MNC ↑	MSE ↓	MNC ↑	MSE ↓	MNC ↑	MSE ↓	MNC ↑
BlindDPS (ours)	**0.003**	**0.955**	**0.000**	**0.995**	**0.003**	**0.930**	**0.001**	**0.991**
SelfDeblur [132]	0.021	0.323	0.020	0.266	0.021	0.268	0.020	0.272
Pan-DCP [122]	0.020	0.425	0.016	0.478	0.020	0.365	0.016	0.481
Pan-ℓ_0 [120]	0.020	0.454	0.016	0.518	0.020	0.398	0.015	0.517

Table 3.6: Quantitative evaluation (MSE, MNC [68]) of kernel estimation on FFHQ and AFHQ. **Bold**: Best, under: second best.

	FFHQ (256 × 256)			ImageNet (256 × 256)		
Method	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑
BlindDPS (ours)	**27.35**	**0.247**	24.49	**51.25**	**0.341**	19.59
TSR-WGAN [73]	58.30	0.258	**26.29**	69.80	0.369	17.67
ILVR [23]	65.50	0.370	21.48	85.21	0.494	18.09
MPRNet [176]	116.2	0.411	19.68	78.24	0.421	20.34
DeblurGANv2 [95]	225.9	0.561	18.40	60.31	0.393	**21.56**

Table 3.7: Quantitative evaluation (FID, LPIPS, PSNR) of imaging through turbulence task on FFHQ and ImageNet. **Bold**: Best, under: second best.

where $\tilde{\boldsymbol{k}}, \boldsymbol{k}^*$ are the estimated, and the ground truth kernels, respectively.

Results

Blind deblurring Motion deblurring results are presented in Fig. 3.6(a) and Fig. 3.9. As our setting for motion deblurring imposes a rather aggressive degradation with a large blur kernel, most of the prior arts fail catastrophically, not being able to generate a feasible solution. In contrast, our method accurately captures both the kernel and the image with sharpness. Similar trend can be seen for Gaussian deblurring presented in the third row of Fig. 3.9. Other methods fall far short of BlindDPS in the sense that they either produce reconstructions that are blurry with inaccurate blur kernel estimation, or fails dramatically (e.g. SelfDeblur). Furthermore, the proposed method establishes the state-of-the-art in all quantitative metrics, which can be seen in Table 3.5 and Table 3.6.

Imaging through turbulence We show the reconstruction results in Fig. 3.6(b) and Fig. 3.10, with quantitative metrics in Table 3.7. Consistent with the results from blind deblurring, BlindDPS outperforms the comparison methods in most cases, effectively removing both the blur and the tilt from the measurement. Notably, our method outperforms *all* other methods by large margins on perceptual metrics (i.e. FID, LPIPS). For PSNR, the proposed method often slightly underperforms against supervised learning approaches, which is to be expected, as for reconstructions from heavy degradations, retrieving the high-frequency details often penalizes such distortion metrics [9].

3.3.2 Ablation studies

We perform two ablation studies to verify our design choices: 1) using the diffusion prior for the forward model parameters, and 2) augmenting the diffusion prior with the sparsity prior.

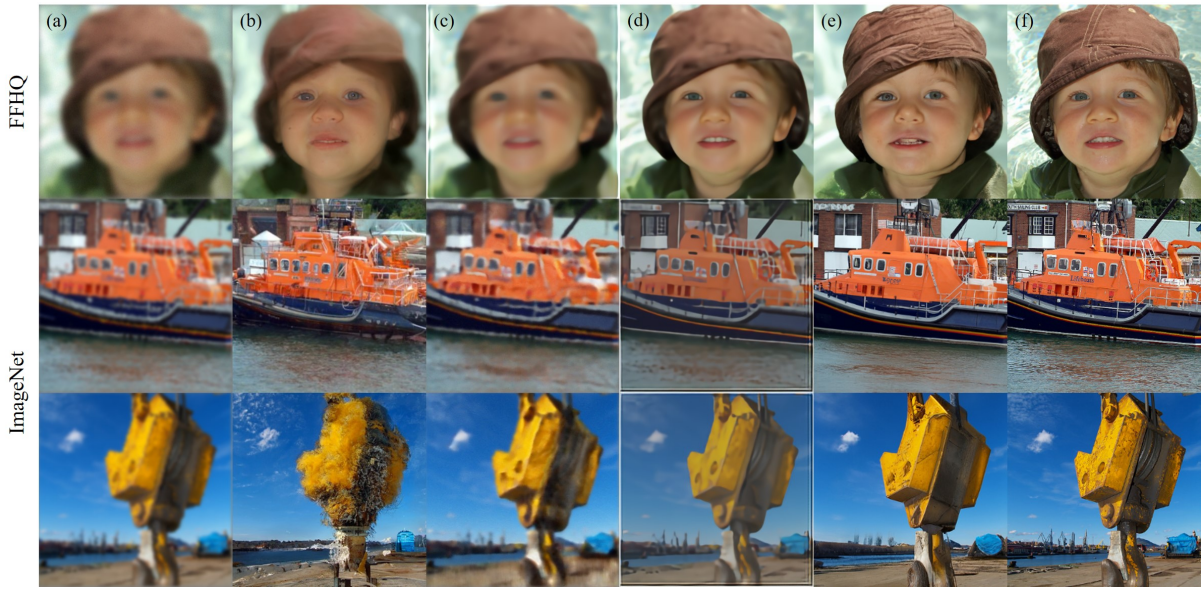


Figure 3.10: Reconstruction of imaging through turbulence. (row 1): FFHQ 256×256, (row 2-3): ImageNet 256×256. (a) Measurement, (b) ILVR [23], (c) MPRNet [176], (d) TSR-WGAN [73], (e) BlindDPS (**ours**), (f) Ground truth.

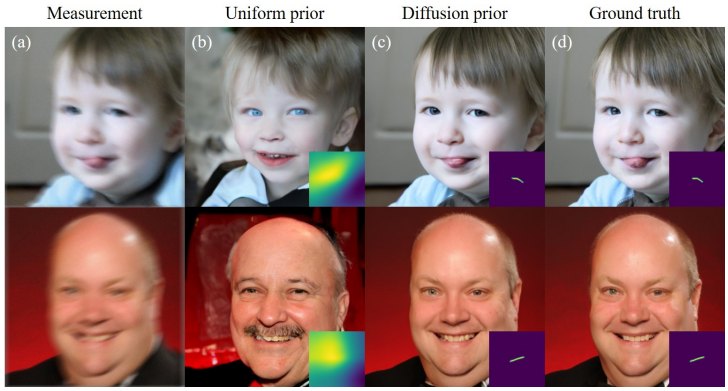


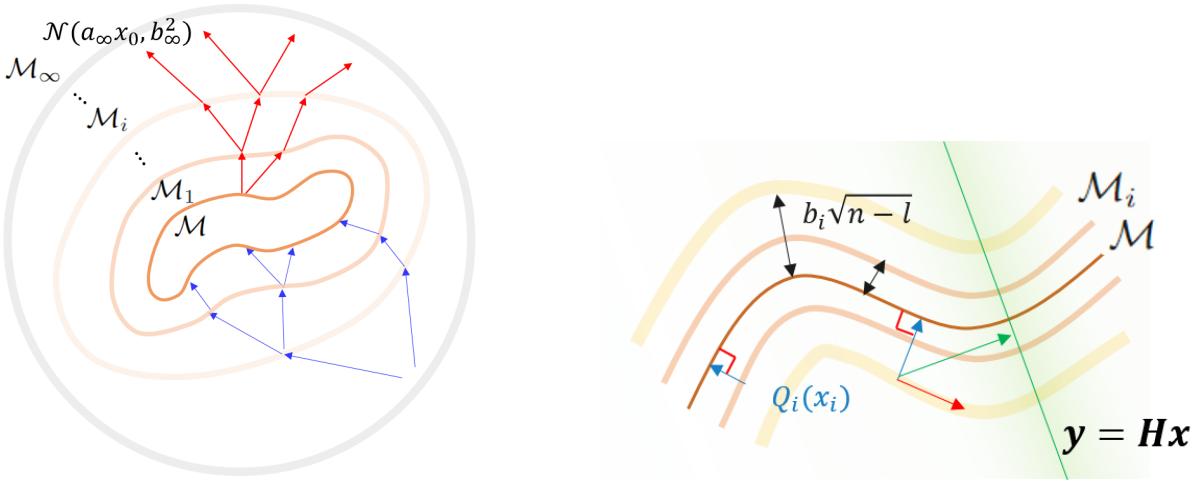
Figure 3.11: Ablation study: uniform prior vs. diffusion prior. (a) Measurement, (b) uniform prior, (c) diffusion prior, (d) ground truth.

Diffusion prior for the forward model One may question why the score function for the kernel is necessary in the first place, since one could also estimate the kernel solely through gradient descent using the gradient of the likelihood. In fact, this corresponds to using the uniform prior for the kernel distribution, which we compare against the proposed diffusion prior (BlindDPS) in Fig. 3.11. We clearly see that using the uniform prior yields heavily distorted results, with a poorly estimated kernel. From this experiment, we observe that using another diffusion process specifically for the forward model is crucial for the performance.

Effect of sparsity regularization One design choice made in BlindDPS is the additional sparsity regularization applied to kernels. Here, we analyze the effect of such regularization. In Table 3.8, we report on quantitative metrics for the kernel, depending on the regularization weight λ. Clearly, setting $\lambda = 0.0$ induces inferior performance especially for motion deblurring. When setting $\lambda \geq 0.1$ however, we can see that one can achieve good performance regardless of the chosen weight value. As diffusion priors have been shown to have surprisingly high generalization capacity [70, 31], we choose a mild weight value of $\lambda = 1.0$, which gives visually appealing results without down-weighting the influence of diffusion priors too much.

	Motion				Gaussian			
λ	0.0	0.1	1.0	5.0	0.0	0.1	1.0	5.0
MNC ↑	0.929	0.956	0.958	0.959	0.996	0.997	0.996	0.997
MSE ↓	0.004	0.002	0.002	0.002	0.000	0.000	0.000	0.000
PSNR ↑	22.43	22.56	22.49	22.60	25.13	25.03	25.00	25.12
FID ↓	81.39	80.25	81.62	82.60	68.48	71.29	72.91	71.86
LPIPS ↓	0.281	0.279	0.281	0.277	0.228	0.230	0.232	0.231

Table 3.8: Ablation study: effect of sparsity regularization in blind deconvolution.



(a) Geometry of diffusion model

(b) MCG correction

Figure 3.12: In both (a) and (b), the central manifolds represent the data manifold $\mathcal{M}$, encircled by manifolds of noisy data $\mathcal{M}_i$. The concentration on the manifold of noisy data and the distance from the clean data manifold are prescribed by Proposition 1. In (a), the backward (resp. forward) step depicted by blue (resp. red) arrows can be considered as transitions from $\mathcal{M}_i$ to $\mathcal{M}_{i-1}$ (resp. $\mathcal{M}_{i-1}$ to $\mathcal{M}_i$). In (b), arrows refer to the directions of conventional projection onto convex sets (POCS) step (green arrow) and MCG step (red arrow) which can be predicted by Theorem 4.

3.4 MCG: Geometric interpretation

In this section, we study the geometric properties of the approximated gradient term proposed in DPS $\nabla_{\boldsymbol{x}_t}\|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0\|_2^2$, which we call the manifold constrained gradient (MCG). To begin with, we borrow a geometrical viewpoint of the data manifold.

Notation and Definitions For a scalar a, points $\boldsymbol{x}, \boldsymbol{y}$ and a set A, we use the following notations. $aA := \{a\boldsymbol{x} : \boldsymbol{x} \in A\}$; $d(\boldsymbol{x}, A) := \inf_{\boldsymbol{y}\in A} ||\boldsymbol{x} - \boldsymbol{y}||_2$; $B_r(A) := \{\boldsymbol{x} : d(\boldsymbol{x}, A) < r\}$; $T_{\boldsymbol{x}}\mathcal{M}$: the tangent space to a manifold $\mathcal{M}$ at $\boldsymbol{x}$; $\boldsymbol{J}_f$: the Jacobian matrix of a vector valued function f. We define $p_0 = p_{data}$. For simplicity in exposition, we consider a linear measurement $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}$. We consider a general forward diffusion perturbation

$$\boldsymbol{x}_i = a_i\boldsymbol{x}_0 + b_i\boldsymbol{z}, \quad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \tag{3.34}$$

Assumption 1 (Strong manifold assumption: linear structure)**.** *Suppose $\mathcal{M} \subset \mathbb{R}^n$ is the set of all data points, here we call the data manifold. Then, the manifold coincides with the tangent space with dimension $l \ll n$.*

$$\mathcal{M} \cap B_R(\boldsymbol{x}_0) = T_{x_0}\mathcal{M} \cap B_R(x_0) \text{ and } T_{x_0}\mathcal{M} \cong \mathbb{R}^l.$$

Moreover, the data distribution p_0 is the uniform distribution on the data manifold $\mathcal{M}$.

Under this assumption, the following proposition shows how the data perturbed by noise lies in the ambient space, illustrated pictorially in Fig. 3.12a.

Proposition 1 (Concentration of noisy data)**.** *Consider the distribution of noisy data $p_i(\boldsymbol{x}_i) = \int p(\boldsymbol{x}_i|\boldsymbol{x})p_0(\boldsymbol{x})d\boldsymbol{x}, p(\boldsymbol{x}_i|\boldsymbol{x}) \sim \mathcal{N}(a_i\boldsymbol{x}, b_i^2\boldsymbol{I})$. Then $p_i(\boldsymbol{x}_i)$ is concentrated on $(n-1)$-dim manifold $\mathcal{M}_i := \{\boldsymbol{y} \in \mathbb{R}^n : d(\boldsymbol{y}, a_i\mathcal{M}) = r_i := b_i\sqrt{n-l}\}$. Rigorously, $p_i(B_{\epsilon r_i}(\mathcal{M}_i)) > 1-\delta$, for some small $\epsilon, \delta > 0$.*

Remark 3 (Geometric interpretation of the diffusion process)**.** *Considering Proposition 1, the manifolds of noisy data can be interpreted as interpolating manifolds between the two: the hypersphere, where pure noise $\mathcal{N}(a_\infty\boldsymbol{x}_0, b_\infty^2)$ is concentrated, and the clean data manifold. In this regard, the diffusion steps are mere transitions from one manifold to another and the diffusion process is a transport from the data manifold to the hypersphere through interpolating manifolds. See Fig. 3.12a.*

Remark 4. *We can infer from the proposition that the score functions are trained only with the data points concentrated on the noisy data manifolds. Therefore, inaccurate inference might be caused by application of a score function on points away from the noisy data manifold.*

Proposition 2 (score function)**.** *Suppose $\boldsymbol{s}_\theta$ is the minimizer of the denoising score matching loss in Eq. (2.17). Let Q_i be the function that maps $\boldsymbol{x}_i$ to $\hat{\boldsymbol{x}}_0$ for each i,*

$$Q_i : \mathbb{R}^d \to \mathbb{R}^d, \boldsymbol{x}_i \mapsto \hat{\boldsymbol{x}}_0 := \frac{1}{a_i}(\boldsymbol{x}_i + b_i^2\boldsymbol{s}_\theta(\boldsymbol{x}_i, i)).$$

Then, $Q_i(\boldsymbol{x}_i) \in \mathcal{M}$ and $\boldsymbol{J}_{Q_i}^2 = \boldsymbol{J}_{Q_i} = \boldsymbol{J}_{Q_i}^T : \mathbb{R}^d \to T_{Q_i(\boldsymbol{x}_i)}\mathcal{M}$. Intuitively, Q_i is locally an orthogonal projection onto $\mathcal{M}$.

According to the proposition, the score function only concerns the normal direction of the data manifold. In other words, the score function cannot discriminate two data points whose difference is tangent to the manifold. In solving inverse problems, however, we desire to discriminate data points to reconstruct the original signal, and the discrimination is achievable by measurement fidelity. In order to achieve the original signal, the measurement plays a role in correcting the tangent component near the data manifold. Furthermore, with regard to remark 4, diffusion model-based inverse problem solvers should follow the tangent component. The following theorem shows how MCG is useful in this regard.

Theorem 4 (Manifold constrained gradient)**.** *A correction by the manifold constrained gradient does not leave the data manifold. Formally,*

$$\frac{\partial}{\partial \boldsymbol{x}_i}\|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0\|_2^2 = -2\boldsymbol{J}_{Q_i}^T\boldsymbol{A}^T(\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0) \in T_{\hat{\boldsymbol{x}}_0}\mathcal{M},$$

the gradient is the projection of the data fidelity term onto $T_{\hat{\boldsymbol{x}}_0}\mathcal{M}$,

This theorem suggests that in diffusion models, a naive measurement fidelity step (without considering the data manifold) pushes the inference path out of the manifolds and might lead to inaccurate reconstruction. On the other hand, MCG guides the diffusion to lie on the data manifold, leading to better reconstruction. Such geometric views are illustrated in Fig. 3.12b.

3.5 Conclusion

In this chapter, we proposed a general framework for solving inverse problems with diffusion models through a Bayesian framework. Using a Bayesian approach, it was shown that one can use a pre-trained diffusion model as a general plug-and-play prior which can be used for various different types of inverse problems without any fine-tuning. In Section 3.2, DPS was proposed for problems where the measurement model was fully known. In Section 3.3, DPS was extended to BlindDPS for the cases where only the functional form of the forward model is known, and the parameters should be estimated. In Section 3.4, a geometric interpretation of solving inverse problems with diffusion models was proposed, showing intriguing properties.

Chapter 4. Decomposed Diffusion Sampler

Two major limitations of DPS (along with other DIS) that were not discussed extensively, are its slow inference time, and the inability to handle 3D inverse problems. As opposed to the usual reconstruction using feed-forward neural networks that are trained by supervised learning, DIS typically involves few thousands of NFE to get a stable result. Even worse, when we consider problems such as DPS, which requires backpropagation through the score function, the problem is even worse, decelerating the inference speed by a factor of 2 or higher. For time-critical large-scale applications such as medical imaging, this is a critical downside that hampers practical deployment. In this chapter, we discuss ways to handle these downsides one by one.

4.1 DiffusionMBIR: 3D inverse problem solving from 2D diffusion

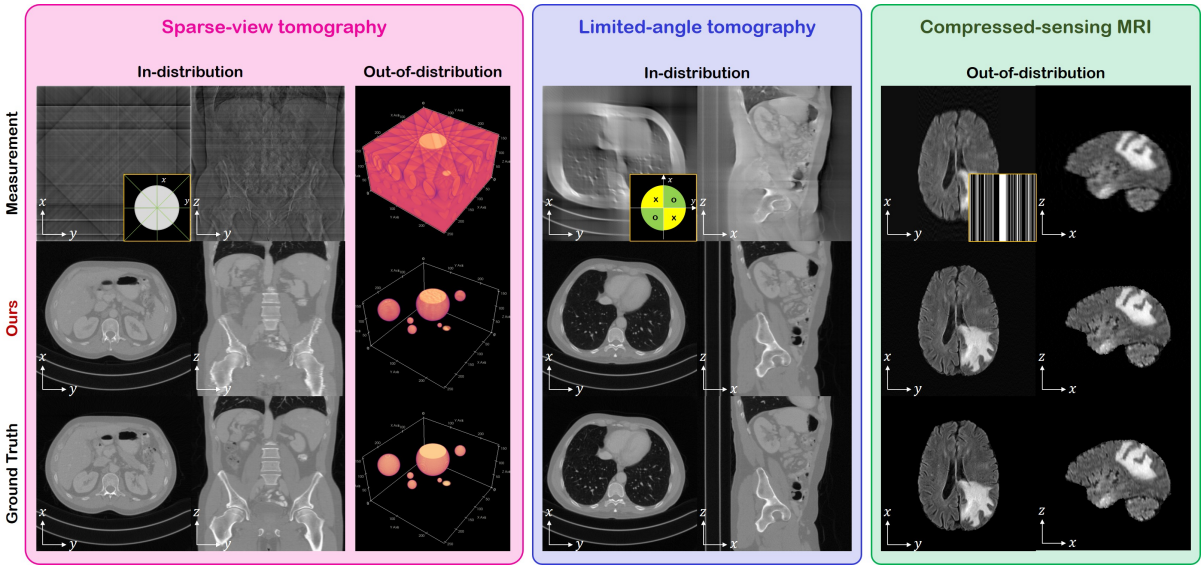


Figure 4.1: 3D reconstruction results with DiffusionMBIR. First row: measurement, second row: our method, third row: ground truth. Yellow inset: measurement process. Sparse-view tomography: 8-view measurement, Limited-angle tomography: [0 90]° out of [0 180]° angle measurement, Compressed-sensing MRI: 1D uniform sub-sampling of $\times 2$ acceleration. (In-distribution): test data aligned with training data, (Out-of-distribution): test data vastly different from training data.

All methods that were considered so far focused on 2D imaging situations. This is mostly due to the high-dimensional nature of the generative constraint. Specifically, diffusion models generate samples by starting from pure noise, and iteratively denoising the data until reaching the clean image. Consequently, the generative process involves staying in the *same* dimension as the data, which is prohibitive when one tries to scale the data dimension to 3D. One should also note that training a 3D diffusion model amounts to learning the 3D prior of the data density. This is undesirable in two aspects. First, the model is data hungry, and hence training a 3D model would typically require thousands of *volumes*, compared to 2D models that could be trained with less than 10 volumes. Second, the prior would be needlessly complicated: when it comes to dynamic imaging or 3D imaging, exploiting the spatial/temporal correlation [158, 76] is standard practice. Naively modeling the problem in 3D would miss the chance to leverage such information.

Another much more well-established method for solving 3D inverse problems is model-based iterative

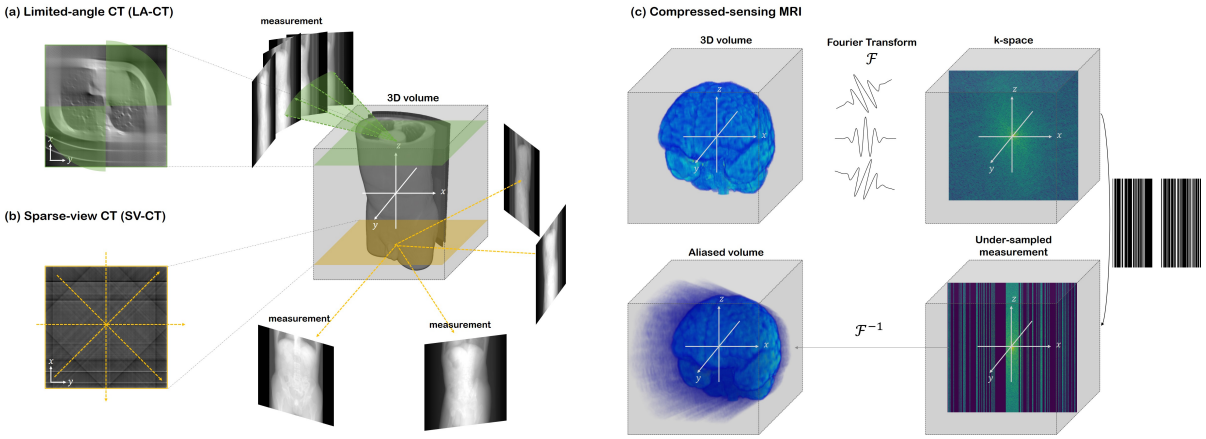


Figure 4.2: Visualization of the measurement process for the three tasks we tackle in this work: (a) Limited angle CT (LA-CT), (b) sparse view CT (SV-CT), (c) compressed sensing MRI (CS-MRI).

reconstruction (MBIR) [81, 108], where the problem is formulated as an optimization problem of weighted least squares (WLS), constructed with the data consistency term, and the regularization term. One of the most widely acknowledged regularizations in the field is the total variation (TV) penalty [146, 109], known for its intriguing properties: edge-preserving while imposing smoothness. While the TV prior has been widely explored, it is known to fall behind the data-driven prior of the modern machine learning practice, as the function is too simplistic to fully model how the image "looks like".

In this section, we propose **DiffusionMBIR**, a method to combine the best of both worlds: we incorporate the MBIR optimization strategy into the diffusion sampling steps in order to *augment* the data-driven prior with the conventional TV prior, imposed to the z-direction only. Particularly, the standard reverse diffusion (i.e. denoising) steps are run independently with respect to the z-axis, and hence standard 2D diffusion models can be used. Subsequently, the data consistency step is imposed by aggregating the slices, then taking a single update step of the alternating direction method of multipliers (ADMM) [11]. This step effectively coerces the cross-talk between the slices with the measurement information, and the TV prior. For efficient optimization, we further propose a strategy which we call *variable sharing*, which enables us to only use a *single* sweep of ADMM and conjugate gradient (CG) per denoising iteration. Note that our method is fully general in that we are not restricted to the given forward operator at test time. Hence, we verify the efficacy of the method by performing extensive experiments on sparse-view CT (SV-CT), limited angle CT (LA-CT), and compressed sensing MRI (CS-MRI): out method shows consistent improvements over the current diffusion model-based inverse problem solvers, and shows strong performance on *all* tasks (For representative results, see Fig. 4.1. For conceptual illustration of the inverse problems, see Fig. 4.2).

In short, the main contributions of this section are to devise a diffusion model-based reconstruction method that 1) operates with the voxel representation, 2) is memory-efficient such that we can scale our solver to much higher dimensions (i.e. $> 256^3$), and 3) is not data hungry, such that it can be trained with less than ten 3D volumes.

Background: model-based iterative reconstruction (MBIR) Consider a linear forward model for an imaging system (e.g. CT, MRI)

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}, \tag{4.1}$$

where $\boldsymbol{y} \in \mathbb{R}^m$ is the measurement (i.e. sinogram, k-space), $\boldsymbol{x} \in \mathbb{R}^n$ is the image that we wish to reconstruct, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is the discrete transform matrix (i.e. Radon, Fourier[1]), and $\boldsymbol{n}$ is the measurement noise in the system. As the problem is ill-posed, a standard approach for the inverse problem that estimates the unknown image $\boldsymbol{x}$ from the measurement $\boldsymbol{y}$ is to perform the following regularized reconstruction:

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + R(\boldsymbol{x}), \tag{4.2}$$

where R is the suitable regularization for $\boldsymbol{x}$, for instance, sparsity in some transformed domain. One widely used function is the TV penalty, $R(\boldsymbol{x}) = \|\boldsymbol{D}\boldsymbol{x}\|_{2,1}$, where $\boldsymbol{D} := [\boldsymbol{D}_x, \boldsymbol{D}_y, \boldsymbol{D}_z]^T$ computes the finite difference in each axis. Minimization of Eq. (4.2) can be performed with robust optimization algorithms, such as fast iterative soft thresholding algorithm (FISTA) [7] or ADMM.

4.1.1 DiffusionMBIR

Main idea To efficiently utilize the diffusion models for 3D reconstruction, one possible solution would be to apply 2D diffusion models slice by slice. However, this approach has one fundamental limitation. When the steps are run without considering the inter-dependency between the slices, the slices that are reconstructed will not be coherent with each other (especially when we have sparser view angles). Consequently, when viewed from the coronal/sagittal slice, the images contain severe artifacts.

In order to address this issue, we are interested in combining the advantages of the MBIR and the diffusion model to oppress unwanted artifacts. Specifically, our proposal is to adopt the alternating minimization approach, but rather than applying it in 2D domain, the diffusion-based denoising step is applied slice-by-slice, whereas the 2-D projection step is replaced with the ADMM update step in 3-D volume. Specifically, we consider the following sub-problem

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \|\boldsymbol{D}_z\boldsymbol{x}\|_1, \tag{4.3}$$

where unlike the conventional TV algorithms that take $\|\boldsymbol{D}\boldsymbol{x}\|_1$, we only take the ℓ_1 norm of the finite difference in the $z-$axis. This choice stems from the fact that the prior with respect to the xy plane is already taken care of with the neural network $\boldsymbol{s}_{\theta^*}$, and all we need to imply is the spatial correlation with respect to the remaining direction. In other words, we are augmenting the generative prior with the model-based sparsity prior. From our experiments, we observe that our prior augmentation strategy is highly effective in producing coherent 3D reconstructions throughout all three axes.

Algorithmic steps We arrive at the update steps

$$\boldsymbol{x}^+ = (\boldsymbol{A}^T\boldsymbol{A} + \rho\boldsymbol{D}_z^T\boldsymbol{D}_z)^{-1}(\boldsymbol{A}^T\boldsymbol{y} + \rho\boldsymbol{D}^T(\boldsymbol{z} - \boldsymbol{w})) \tag{4.4}$$

$$\boldsymbol{z}^+ = \mathcal{S}_{\lambda/\rho}(\boldsymbol{D}_z\boldsymbol{x}^+ + \boldsymbol{w}) \tag{4.5}$$

$$\boldsymbol{w}^+ = \boldsymbol{w} + \boldsymbol{D}_z\boldsymbol{x}^+ - \boldsymbol{z}^+, \tag{4.6}$$

where ρ is the hyper-paremeter for the method of multipliers, and $\mathcal{S}$ is the soft thresholding operator. Moreover, Eq. (4.4) can be solved with conjugate gradient (CG), which efficiently finds a solution for $\boldsymbol{x}$ that satisfies $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$: we denote running K iterations of CG with initial point $\boldsymbol{x}$ as $\texttt{CG}(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{x}, K)$. Full derivation for the ADMM steps

[1]While we denote real-valued transforms and measurements for the simplicity of exposition, the discrete Fourier transform (DFT) matrix and the corresponding measurement are complex-valued.

Algorithm 4 DiffusionMBIR concept

Require: s_{θ^*}, N

$\boldsymbol{x}_N \sim \mathcal{N}(\mathbf{0}, \sigma_T^2 \boldsymbol{I})$
for $i = N-1:0$ **do**
  $\boldsymbol{x}_i' \leftarrow \text{Denoise}(\boldsymbol{x}_{i+1}, s_{\theta^*})$
  $\boldsymbol{x}_i \leftarrow \arg\min_{\boldsymbol{x}_i'} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_i'\|_2^2 + \|\boldsymbol{D}_z \boldsymbol{x}_i'\|_1$
end for
return $\boldsymbol{x}_0$

Algorithm 5 DiffusionMBIR (fast; variable sharing)

Require: $s_{\theta^*}, N, \lambda, \rho, \{\sigma_i\}$

$\boldsymbol{x}_N \sim \mathcal{N}(\mathbf{0}, \sigma_T^2 \boldsymbol{I})$
$\boldsymbol{z}_N \leftarrow \texttt{torch.zeros\_like}(\boldsymbol{x}_N)$
$\boldsymbol{w}_N \leftarrow \texttt{torch.zeros\_like}(\boldsymbol{x}_N)$
for $i = N-1:0$ **do**
  $\boldsymbol{x}_i' \leftarrow \text{Denoise}(\boldsymbol{x}_{i+1}, s_{\theta^*})$
  $\boldsymbol{A}_{\text{CG}} \leftarrow \boldsymbol{A}^T\boldsymbol{A} + \rho \boldsymbol{D}_z^T \boldsymbol{D}_z$
  $\boldsymbol{b}_{\text{CG}} \leftarrow \boldsymbol{A}^T\boldsymbol{y} + \rho \boldsymbol{D}_z^T(\boldsymbol{z}_{i+1} - \boldsymbol{w}_{i+1})$
  $\boldsymbol{x}_i \leftarrow \texttt{CG}(\boldsymbol{A}_{\text{CG}}, \boldsymbol{b}_{\text{CG}}, \boldsymbol{x}_i', 1)$
  $\boldsymbol{z}_i \leftarrow \mathcal{S}_{\lambda/\rho}(\boldsymbol{D}_z \boldsymbol{x}_i + \boldsymbol{w}_{i+1})$
  $\boldsymbol{w}_i \leftarrow \boldsymbol{w}_{i+1} + \boldsymbol{D}_z \boldsymbol{x}_i - \boldsymbol{z}_i$
end for
return $\boldsymbol{x}_0$

	8-view						4-view						2-view					
	Axial*		Coronal		Sagittal		Axial*		Coronal		Sagittal		Axial*		Coronal		Sagittal	
Method	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
DiffusionMBIR (ours)	**33.49**	**0.942**	**35.18**	**0.967**	**32.18**	**0.910**	**30.52**	**0.914**	**30.09**	**0.938**	**27.89**	**0.871**	24.11	0.810	23.15	**0.841**	**21.72**	**0.766**
Chung *et al.* [29]	28.61	0.873	28.05	0.884	24.45	0.765	27.33	0.855	26.52	0.863	23.04	0.745	**24.69**	**0.821**	**23.52**	0.806	20.71	0.685
Lahiri *et al.* [97]	21.38	0.711	23.89	0.769	20.81	0.716	20.37	0.652	21.41	0.721	18.40	0.665	19.74	0.631	19.92	0.720	17.34	0.650
FBPConvNet [74]	16.57	0.553	19.12	0.774	18.11	0.714	16.45	0.529	19.47	0.713	15.48	0.610	16.31	0.521	17.05	0.521	11.07	0.483
ADMM-TV	16.79	0.645	18.95	0.772	17.27	0.716	13.59	0.618	15.23	0.682	14.60	0.638	10.28	0.409	13.77	0.616	11.49	0.553

Table 4.1: Quantitative evaluation of SV-CT (8, 4, 2-view) (PSNR, SSIM) on the AAPM 256×256 test set. **Bold**: Best, under: second best.*: the plane where the diffusion model prior takes place.

is provided in Supplementary section A.2.1. For simplicity, we denote one sweep of Eq. (4.4),Eq. (4.5),Eq. (4.6) as $\boldsymbol{x}^+, \boldsymbol{z}^+, \boldsymbol{w}^+ = \texttt{ADMM}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{w})$. Iterative application of $\texttt{ADMM}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{w})$ would robustly solve the minimization problem in Eq. (4.3). Hence, the naive implementation of the proposed algorithm would be

$$\boldsymbol{x}_{i-1}' \leftarrow \text{Solve}(\boldsymbol{x}_i, \boldsymbol{s}_{\theta^*}), \tag{4.7}$$

$$\boldsymbol{x}_{i-1} \leftarrow \arg\min_{\boldsymbol{x}_{i-1}'} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_{i-1}'\|_2^2 + \|\boldsymbol{D}_z \boldsymbol{x}_{i-1}'\|_1. \tag{4.8}$$

Specifically, Eq. (4.7) would amount to parallel denoising for each slice, whereas Eq. (4.8) augments the z-directional TV prior and imposes consistency. Here, note that there are three sources of iteration in the algorithm: 1) Numerical integration of SDE, indexed with i, 2) ADMM iteration, and 3) the inner CG iteration, used to solve Eq. (4.4). Since diffusion models are slow in themselves, the multiplicative additional cost of factors 2),3) will be prohibitive, and should be refrained from. In the following, we devise a simple method to reduce this cost dramatically.

Fast and efficient implementation (variable sharing) Naively implementing Algorithm 4, we would re-initialize the primal variable $\boldsymbol{z}$, and the dual variable $\boldsymbol{w}$, every time before the ADMM iteration runs for the i^{th} iteration

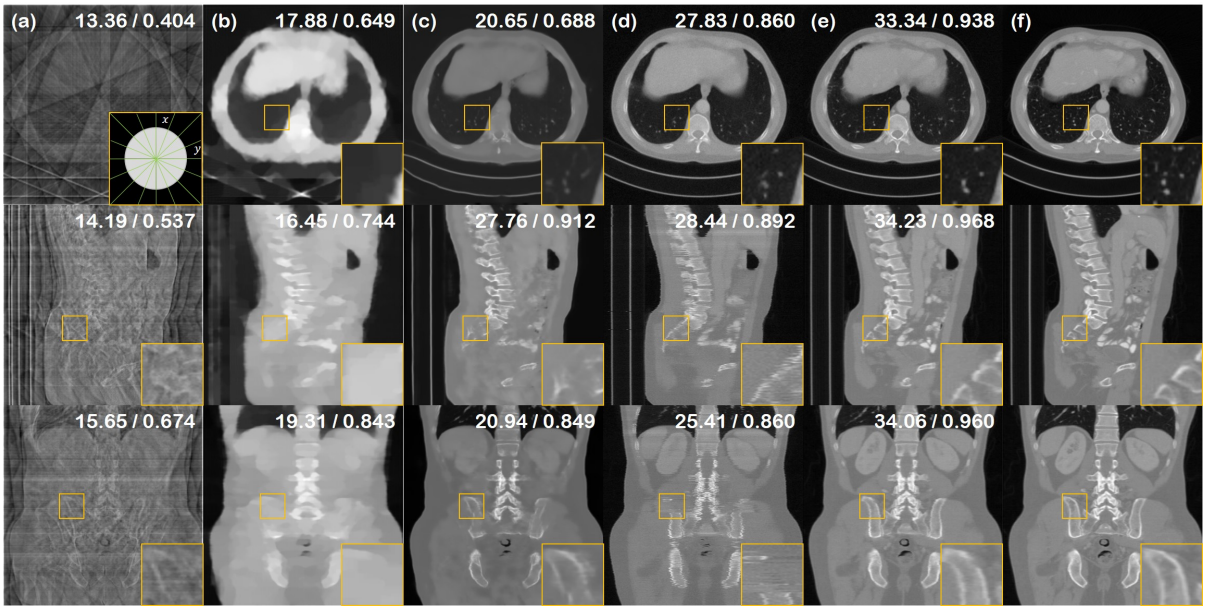


Figure 4.3: 8-view SV-CT reconstruction results of the test data (First row: axial slice, second row: sagittal slice, third row: coronal slice). (a) FBP, (b) ADMM-TV, (c) Lahiri *et al.* [97], (d) Chung *et al.* [29], (e) proposed method, (f) ground truth. PSNR/SSIM values are presented in the upper right corner. Green lines in the inset of the first row (a): measured angles.

of the SDE. In turn, this would lead to slow convergence of the ADMM algorithm, as the burn-in period for the variables $\boldsymbol{z}, \boldsymbol{w}$ would be required for the first few iterations. Moreover, since solving for diffusion models would have a large number of discretization steps N, the difference between the two adjacent iterations $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i+1}$ is minimal. When dropping the values of $\boldsymbol{z}, \boldsymbol{w}$ from the $i+1^{\text{th}}$ iteration and re-initializing at the i^{th} iteration, one would be dropping valuable information, and wasting compute. Hence, we propose to initialize both $\boldsymbol{z}_N, \boldsymbol{w}_N$ as a *global* variable before the start of the SDE iteration, and keep the updated values throughout. Interestingly enough, we find that choosing $M = 1, K = 1$, i.e. *single* iteration for both ADMM and CG is necessary for high-fidelity reconstruction. Our fast version of DiffusionMBIR is presented in Algorithm 5.

Another caveat is that running the neural network forward pass through the entire volume is not feasible memory-wise, for example, when fitting the solver into a single commodity GPU. One can circumvent this by dividing the batch dimension[2] into sub-batches, running the denoising step for the sub-patches separately, and then aggregating them into the full volume again. The `ADMM` step can be applied to the full volume after the aggregation, which would yield the same solution with Algorithm 5. For both the slow and the fast version of the algorithm, one can also apply a projection to the measurement subspace at the end when one wishes to exactly match the measurement constraint.

Experiments

We conduct experiments on the three most widely studied tasks in medical image reconstruction: 1) sparse view CT (SV-CT), 2) limited angle CT (LA-CT), and 3) compressed sensing MRI (CS-MRI). For both CT reconstruction tasks (i.e. SV-CT, LA-CT) we use the data from the AAPM 2016 CT low-dose grand challenge. All volumes except for one are used for training the 2D score function and one volume is held out for testing. For the task of CS-MRI, we take the data from the multimodal brain tumor image segmentation benchmark (BRATS) [114] 2018 FLAIR volume for testing. Note that we use a pre-trained score function that was trained on fastMRI knee [177] images

[2] In our implementation, the batch dimension corresponds to the $z-$axis, as 2D slices are stacked.

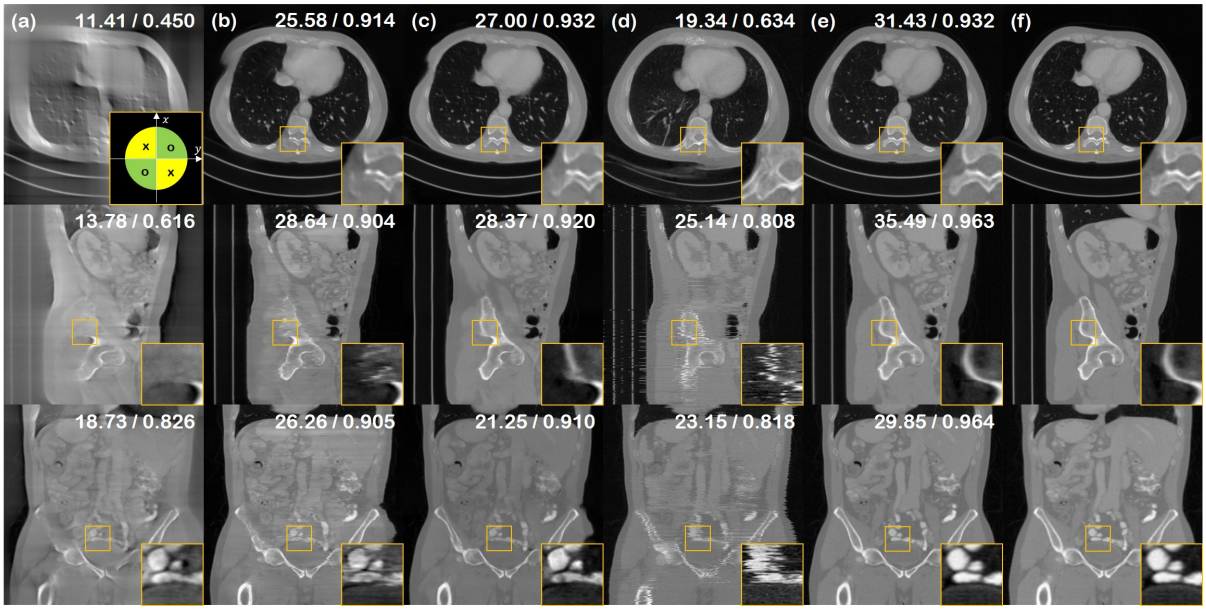


Figure 4.4: 90° LA-CT reconstruction results of the test data (First row: axial slice, second row: sagittal slice, third row: coronal slice). (a) FBP, (b) Zhang *et al.* [178], (c) Lahiri *et al.* [97], (d) Chung *et al.* [29], (e) proposed method, (f) ground truth. PSNR/SSIM values are presented in the upper right corner. The green area in the inset of first row (a): measured, Yellow area in the inset of first row (a): not measured.

only, and hence we need not split the train/test data here.

For CT tasks, we train the `ncsnpp` model [156] on the AAPM dataset which consists of about 3000 2D slices of training data. For the CS-MRI task, we take the pre-trained model from score-MRI. For inference (i.e. generation; inverse problem solving), we base our sampler on the predictor-corrector (PC) sampling scheme of [156]. We set $N = 2000$, which amounts to 4000 iterations of NFE with $\boldsymbol{s}_{\theta^*}$.

Results We present the quantitative metrics of the SV-CT reconstruction results in Table 4.1. The table shows that the proposed method outscores the baselines by large margins in most of the settings. Fig. 4.3. As shown in the first row of each figure, axial slices of the proposed method have restored much finer details compared to the baselines. Furthermore, the results of sagittal and coronal slices in the second and third rows imply that DiffusionMBIR could maintain the structural connectivity of the original structures in all directions. In contrast, Chung et al. [29] perform well on reconstructing the axial slices, but do not have spatial integrity across the z direction, leading to shaggy artifacts that can be clearly seen in coronal/sagittal slices. Lahiri et al. [97] often omits important details, and is not capable of reconstruction, especially when we only have 4 number of views. ADMM-TV hardly produces satisfactory results due to the extremely limited setting.

The results of the limited angle tomography are presented in Fig. 4.4. We test on the case where we have measurements in the $[0, 90]^\circ$ regime, and no measurements in the $[90, 180]^\circ$ regime. Hence, the task is to *infill* the missing views. Consistent with what was observed from SV-CT experiments, we see that DiffusionMBIR improves over the conventional diffusion model-based method [29], and also outperforms other fully supervised methods, where we see even larger gaps in performance between the proposed method and all the other methods. Notably, Chung et al. [29] leverages no information from the adjacent slices, and hence has a high degree of freedom on how to infill the missing angle. As the reconstruction is stochastic, we cannot impose consistency across the different slices. Often, this results in the structure of the torso being completely distorted, as can be seen in the first row of Fig. 4.4 (d). In contrast, our augmented prior imposes smoothness across frames, and also naturally robustly preserves the structure.

4.2 DDS: Fast sampling using Krylov subspace methods

One of the most widely acknowledged accelerated diffusion sampling strategies is the denoising diffusion implicit model (DDIM) [151], where the stochastic ancestral sampling of denoising diffusion probabilistic models (DDPM) can be transitioned to deterministic sampling, and thereby accelerate the sampling. Accordingly, DDIM sampling has been well incorporated in solving low-level vision inverse problems [82, 152]. In a recent application of DDIM for linear image restoration tasks, [171] proposed an algorithm dubbed denoising diffusion null-space models (DDNM), where one-step null-space modification is made to impose consistency. However, the sampling strategy is not successful in practical large-scale medical imaging contexts when the forward model is significantly more complex (e.g. parallel imaging (PI) CS-MRI, 3D modalities). Furthermore, it is unclear how the algorithm is related to the existing literature of conditional sampling approaches that take into account the geometry of the manifold, considered in Sec. 3.2.

On the other hand, in classical optimization literature, Krylov subspace methods have been extensively studied to deal with large-scale inverse problems due to their rapid convergence rates [102]. Specifically, consider a typical linear inverse problem

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, \tag{4.9}$$

where $\boldsymbol{A}$ is the linear mapping and the goal is to retrieve $\boldsymbol{x}$ from the measurement $\boldsymbol{y}$. Without loss of generality, we assume that $\boldsymbol{A}$ in Eq. (4.9) is square. Otherwise, we can obtain an equivalent inverse problem with symmetric linear mapping $\tilde{\boldsymbol{A}}$ as $\tilde{\boldsymbol{y}} := \boldsymbol{A}^*\boldsymbol{y} = \boldsymbol{A}^*\boldsymbol{A}\boldsymbol{x} := \tilde{\boldsymbol{A}}\boldsymbol{x}$. This is because the solution to the normal equation $\boldsymbol{A}^*\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^*\boldsymbol{y}$ is indeed a solution to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$ if $\boldsymbol{A}^*$ has full column rank, which holds in most of the ill-posed inverse problem cases. Given an initial guess $\hat{\boldsymbol{x}}$, Krylov subspace methods seek an approximate solution $\boldsymbol{x}^{(l)}$ from an affine subspace $\hat{\boldsymbol{x}} + \mathcal{K}_l$, where the l-th order Krylov subspace $\mathcal{K}_l$ is defined by

$$\mathcal{K}_l := \mathrm{Span}(\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \cdots, \boldsymbol{A}^{l-1}\boldsymbol{b}), \quad \boldsymbol{b} := \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}} \tag{4.10}$$

For example, the conjugate gradient (CG) method is a special class of the Krylov subspace method that minimizes the residual in the Krylov subspace. Krylov subspace methods are particularly useful for large-scale problems thanks to their fast convergence [102].

Inspired by this, here we are interested in developing a method that synergistically combines Krylov subspace methods with diffusion models such that it can be effectively used for large-scale inverse problems. Specifically, based on a novel observation that a diffusion posterior sampling (DPS) [26] with the manifold constrained gradient (MCG) [29] is equivalent to one-step projected gradient on the tangent space at the "denoised" data by Tweedie's formula, we provide *multi-step* update scheme on the tangent space using Krylov subspace methods. Specifically, we show that the multiple CG updates are guaranteed to remain in the tangent space, and subsequently generated sample with the addition of the noise component can be correctly transferred to the next noisy manifold. This eliminates the need for the computationally demanding MCG while permitting multiple *economical* CG steps at each ancestral diffusion sampling, resulting in a more efficient DDIM sampling. Our analysis holds for both variance-preserving (VP) and variance-exploding (VE) sampling schemes.

The combined strategy, dubbed **D**ecomposed **D**iffusion **S**ampling (DDS), yields *better* performance with much-reduced sampling time (20$\sim$50 NFE, $\times 80 \sim 200$ acceleration; See Fig. 4.5 for representative results), and is shown to be applicable to a variety of challenging large scale inverse problem tasks: multi-coil MRI reconstruction and 3D CT reconstruction.

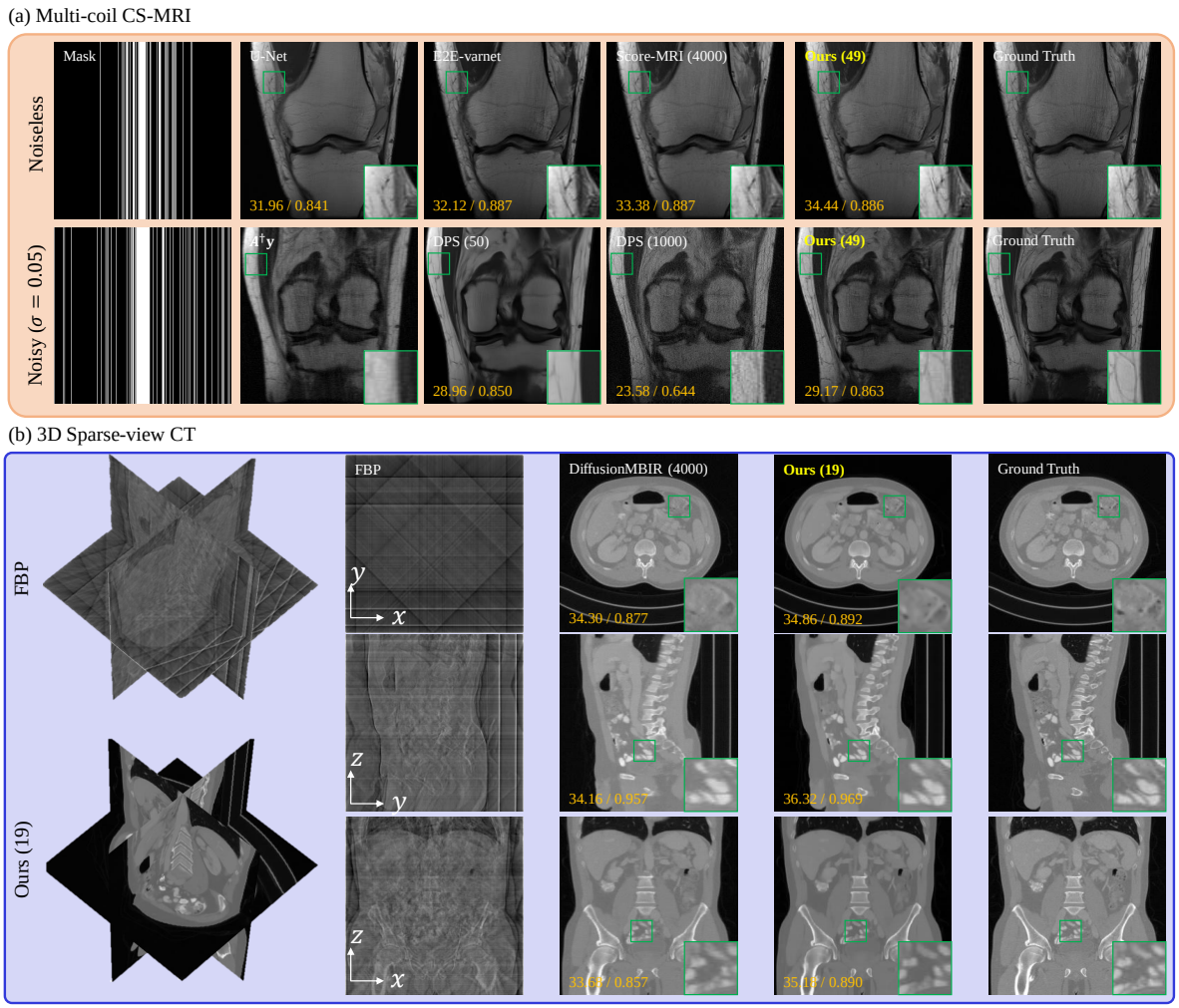


Figure 4.5: Representative reconstruction results. (a) Multi-coil MRI reconstruction, (b) 3D sparse-view CT. Numbers in parenthesis: NFE. Yellow numbers in bottom left corner: PSNR/SSIM.

Conditional diffusion for inverse problems The conditional diffusion sampling for inverse problems [82, 26, 28] attempts to solve the following optimization problem:

$$\min_{\boldsymbol{x}\in\mathcal{M}} \ell(\boldsymbol{x}) \tag{4.11}$$

where $\ell(\boldsymbol{x})$ denotes the data consistency (DC) loss (i.e., $\ell(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2/2$ for linear inverse problems) and $\mathcal{M}$ represents the clean data manifold. Consequently, it is essential to navigate in a way that minimizes cost while also identifying the correct clean manifold. Accordingly, most of the approaches use standard reverse diffusion, alternated with an operation to minimize the DC loss.

Recently, [26] proposed DPS, where the updated estimate from the noisy sample $\boldsymbol{x}_t \in \mathcal{M}_t$ is constrained to stay on the same noisy manifold $\mathcal{M}_t$. This is achieved by computing the MCG [29] on a noisy sample $\boldsymbol{x}_t \in \mathcal{M}_t$ as $\nabla^{mcg}_{\boldsymbol{x}_t}\ell(\boldsymbol{x}_t) := \nabla_{\boldsymbol{x}_t}\ell(\hat{\boldsymbol{x}}_t)$, where $\hat{\boldsymbol{x}}_t$ is the denoised sample through Tweedie's theorem. The resulting algorithm can be stated as follows:

$$\boldsymbol{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\hat{\boldsymbol{x}}_t - \gamma_t \nabla_{\boldsymbol{x}_t}\ell(\hat{\boldsymbol{x}}_t)\right) + \tilde{\boldsymbol{w}}b_t \tag{4.12}$$

where $\gamma_t > 0$ denotes the step size. Since parameterized score function $\boldsymbol{\epsilon}^{(t)}_{\theta^*}(\boldsymbol{x}_t)$ is trained with samples supported

on $\mathcal{M}_t$, $\boldsymbol{\epsilon}_{\theta^*}^{(t)}$ shows good performance on denoising $\boldsymbol{x}_t \sim \mathcal{M}_t$, allowing precise transition to $\mathcal{M}_{t-1}$. Therefore, by performing DDIM steps from $t = T$ to $t = 0$, we can solve the optimization problem Eq. (4.11) with $\boldsymbol{x}_0 \in \mathcal{M}$. Unfortunately, the computation of MCG for DPS requires computationally expensive backpropagation and is often unstable [128, 41].

Key observation By applying the chain rule for the MCG term in Eq. (6.21), we have

$$\nabla_{\boldsymbol{x}_t} \ell(\hat{\boldsymbol{x}}_t) = \frac{\partial \hat{\boldsymbol{x}}_t}{\partial \boldsymbol{x}_t} \nabla_{\hat{\boldsymbol{x}}_t} \ell(\hat{\boldsymbol{x}}_t)$$

where we use the denominator layout for vector calculus. Since $\nabla_{\hat{\boldsymbol{x}}_t} \ell(\hat{\boldsymbol{x}}_t)$ is a standard gradient, the main complexity of the MCG arises from the Jacobian term $\frac{\partial \hat{\boldsymbol{x}}_t}{\partial \boldsymbol{x}_t}$.

In the following Proposition 3, we show that if the underlying clean manifold forms an affine subspace, then the Jacobian term $\frac{\partial \hat{\boldsymbol{x}}_t}{\partial \boldsymbol{x}_t}$ is indeed the orthogonal projection on the clean manifold up to a scale factor. Note that the affine subspace assumption is widely used in the diffusion literature that has been used when 1) studying the possibility of score estimation and distribution recovery [21], 2) showing the possibility of signal recovery [137, 139], and the most relevantly, 3) showing the geometrical view of the clean and the noisy manifolds [29]. Although it is difficult to assume in practice that the clean manifold forms an affine subspace, it could be approximated by piece-wise linear regions represented by the tangent subspace at $\hat{\boldsymbol{x}}_t$. Therefore, Proposition 3 is still valid in that approximate regime.

Proposition 3 (Manifold Constrained Gradient)**.** *Suppose the clean data manifold $\mathcal{M}$ is represented as an affine subspace and assumes the uniform distribution on $\mathcal{M}$. Then,*

$$\frac{\partial \hat{\boldsymbol{x}}_t}{\partial \boldsymbol{x}_t} = \frac{1}{\sqrt{\bar{\alpha}_t}} \mathcal{P}_{\mathcal{M}} \tag{4.13}$$

$$\hat{\boldsymbol{x}}_t - \gamma_t \nabla_{\boldsymbol{x}_t} \ell(\hat{\boldsymbol{x}}_t) = \mathcal{P}_{\mathcal{M}} \left(\hat{\boldsymbol{x}}_t - \zeta_t \nabla_{\hat{\boldsymbol{x}}_t} \ell(\hat{\boldsymbol{x}}_t) \right) \tag{4.14}$$

for some $\zeta_t > 0$, where $\mathcal{P}_{\mathcal{M}}$ denotes the orthogonal projection to $\mathcal{M}$.

Now, Eq. (4.14) in Proposition 3 indicates that if the clean data manifold is an affine subspace, the DPS corresponds to the projected gradient on the clean manifold. Nonetheless, a notable limitation of MCG is its inefficient use of a single projected gradient step for each ancestral diffusion sampling. Motivated by this, we aim to explore extensions that allow computationally efficient multi-step optimization steps per each ancestral sampling.

Specifically, let $\mathcal{T}_t$ denote the tangent space on the clean manifold at a denoised sample $\hat{\boldsymbol{x}}_t$. Suppose, furthermore, that there exists the l-th order Krylov subspace:

$$\mathcal{K}_{t,l} := \text{Span}(\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \cdots, \boldsymbol{A}^{l-1}\boldsymbol{b}), \quad \boldsymbol{b} := \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_t \tag{4.15}$$

such that

$$\mathcal{T}_t = \hat{\boldsymbol{x}}_t + \mathcal{K}_{t,l}$$

Then, using the property of CG in Eq. (A.62), it is easy to see that M-step CG update with $M \leq l$ starting from $\hat{\boldsymbol{x}}_t$ are confined in $\mathcal{T}_t$ since it corresponds to the solution of

$$\min_{\boldsymbol{x} \in \hat{\boldsymbol{x}}_t + \mathcal{K}_M} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 \tag{4.16}$$

and $\mathcal{K}_M \subset \mathcal{K}_l$ when $M \leq l$. This offers a pivotal insight. It shows that if the tangent space at each denoised sample is representable by a Krylov subspace, there's no need to compute the MCG. Rather, the standard CG method

suffices to guarantee that the updated samples stay within the tangent space. To sum up, our DDS algorithm is as follows:

$$\boldsymbol{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{x}}_t' + \tilde{\boldsymbol{w}}b_t, \tag{4.17}$$

$$\hat{\boldsymbol{x}}_t' = \texttt{CG}(\boldsymbol{A}^*\boldsymbol{A}, \boldsymbol{A}^*\boldsymbol{y}, \hat{\boldsymbol{x}}_t, M), \quad M \leq l \tag{4.18}$$

where $\texttt{CG}(\cdot)$ denotes the M-step CG for the normal equation starting from $\hat{\boldsymbol{x}}_t$. In contrast, DDNM [171] for noiseless image restoration problems uses the following update instead of Eq. (4.18):

$$\hat{\boldsymbol{x}}_t' = (\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A})\hat{\boldsymbol{x}}_t + \boldsymbol{A}^\dagger\boldsymbol{y}, \tag{4.19}$$

where $\boldsymbol{A}^\dagger$ denotes the pseudo-inverse of $\boldsymbol{A}$. Unfortunately, Eq. (4.19) in DDNM does not ensure that the update signal $\hat{\boldsymbol{x}}_t'$ lies in $\mathcal{T}_t$ due to $\boldsymbol{A}^\dagger$.

Therefore, for large-scale inverse problems, we find that CG outperforms naive projections Eq. (4.19) by quite large margins. This is to be expected, as CG iteration enforces the update to stay on $\mathcal{T}_t$ whereas the orthogonal projections in DDNM do not guarantee this property. In practice, even when our Krylov subspace assumptions cannot be guaranteed, we empirically validate that DDS indeed keeps $\boldsymbol{x}_t$ closest to the noisy manifold $\mathcal{M}_t$, which, in turn, shows that DDS keeps the update close to the clean manifold $\mathcal{M}$.

For this experiment, we take 50 random proton density (PD) weighted images from the fastMRI validation dataset, and add Gaussian noise $\sigma_{\text{GT}} = 7.00[\times 10^{-2}]$ to the images. For each noisy image, we apply the following DC step for each method

1. Score-MRI [31]
2. Jalal *et al.* [70]: step size as used in the implementation[3]
3. DPS [26]: step size 1.0
4. DDNM [171]: Eq. (4.19)
5. DDS (CG): CG applied to the Tweedie denoised estimate, $M = 5$.

Once the update step is performed, the Gaussian noise level of the updated samples is estimated with the method from [18]. Note that as the estimation method is imperfect, there is already a gap between the ground truth noise level and the estimated noise level.

Method	No process	Score-MRI	Jalal *et al.*	DPS	DDNM	**DDS (CG)**
σ_{est}	7.556	5.959	6.303	8.527	8.256	7.859
$\lvert\sigma_{\text{est}}^{\text{np}} - \sigma_{\text{est}}\rvert$	0.000	1.597	1.253	0.917	0.700	**0.303**

Table 4.2: Noise offset experiment. Gaussian noise level estimated with [18]. Real noise level: $\sigma_{\text{GT}} = 7.00[\times 10^{-2}]$; $\sigma_{\text{est}}^{\text{np}} = 7.56[\times 10^{-2}]$

Moreover, it is worth emphasizing that gradient-based methods [70, 29, 26] often fail when choosing the "theoretically correct" step sizes of the likelihood. To fix this, several heuristics on the choice of step sizes are required (e.g. choosing the step size $\propto 1/\|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_{\text{t}}\|_2$), which easily breaks when varying the NFE. In this regard, DDS is beneficial in that it is free from the cumbersome step-size tuning process.

[3] https://github.com/utcsilab/csgm-mri-langevin/blob/main/main.py

Furthermore, our CG step can be easily extended for noisy image restoration problems. Unlike the DDNM approach that relies on the singular value decomposition to handle noise, which is non-trivial to perform on forward operators in medical imaging (e.g. PI CS-MRI, CT), we can simply minimize the cost function

$$\ell(\boldsymbol{x}) = \frac{\gamma}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \frac{1}{2}\|\boldsymbol{x} - \hat{\boldsymbol{x}}_t\|_2^2, \tag{4.20}$$

by performing CG iteration $\mathtt{CG}(\gamma \boldsymbol{A}^*\boldsymbol{A} + \boldsymbol{I}, \hat{\boldsymbol{x}}_t + \gamma \boldsymbol{A}^*\boldsymbol{y}, \hat{\boldsymbol{x}}_t, M)$ in the place of Eq. (4.18), where γ is a hyper-parameter that weights the proximal regularization [123]. Finally, our method can also be readily extended to accelerate DiffusionMBIR [28] for 3D CT reconstruction by adhering to the same principles. Specifically, we implement an optimization strategy to impose the conditioning:

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{D}_z\boldsymbol{x}\|_1, \tag{4.21}$$

where $\boldsymbol{D}_z$ is the finite difference operator that is applied to the z-axis that is not learned through the diffusion prior, and unlike [28], the optimization is performed in the clean manifold starting from the denoised $\hat{\boldsymbol{x}}_t$ rather than the noisy manifold starting from $\boldsymbol{x}_t$. As the additional prior is only imposed in the direction that is orthogonal to the axial slice dimension (xy) captured by the diffusion prior (i.e. manifold $\mathcal{M}$), Eq. (4.21) can be solved effectively with the alternating direction method of multipliers (ADMM) [11] after sampling 2D diffusion slice by slice.

4.2.1 Experiments

Improvement over DDNM Fixing the sampling strategy the same, we inspect the effect of the three different data consistency imposing strategies: Score-MRI [31], DDNM [171], and DDS. For DDS, we additionally search for the optimal number of CG iterations per sampling step. We see that under the low NFE regime, the score-MRI DC strategy has significantly worse performance than the proposed methods, even when using the same DDIM sampling strategy. Moreover, we see that overall, DDS outperforms DDNM by a few db in PSNR. We see that 5 CG iterations per denoising step strike a good balance. One might question the additional computational overhead of introducing the iterative CG into the already slow diffusion sampling. Nonetheless, from our experiments, we see that on average, a single CG iteration takes about 0.004 sec. Consequently, it only takes about 0.2 sec more than the analytic counterpart when using 50 NFE (Analytic: 4.51 sec vs. CG(5): 4.71 sec.).

Algorithm 6 DDS (PI MRI; VP; noisy)

Require: $\boldsymbol{\epsilon}_{\theta^*}, N, \{\alpha_t\}_{t=1}^N, \eta, \boldsymbol{A}, M, \gamma$
$\boldsymbol{x}_N \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
for $t = N : 2$ **do**
  $\hat{\boldsymbol{\epsilon}}_t \leftarrow \boldsymbol{\epsilon}_{\theta^*}(\boldsymbol{x}_t)$
  $\hat{\boldsymbol{x}}_t \leftarrow (\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}_t)/\sqrt{\bar{\alpha}_t}$
  $\boldsymbol{A}_{\mathrm{CG}} \leftarrow \boldsymbol{I} + \gamma \boldsymbol{A}^*\boldsymbol{A}$
  $\boldsymbol{y}_{\mathrm{CG}} \leftarrow \hat{\boldsymbol{x}}_t + \gamma \boldsymbol{A}^*\boldsymbol{y}$
  $\hat{\boldsymbol{x}}_t' \leftarrow \mathtt{CG}(\boldsymbol{A}_{\mathrm{CG}}, \boldsymbol{y}_{\mathrm{CG}}, \hat{\boldsymbol{x}}_t, M)$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \mathbf{I})$
  $\boldsymbol{x}_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{x}}_t' - \sqrt{1 - \bar{\alpha}_{t-1} - \eta^2\tilde{\beta}_t^2}\hat{\boldsymbol{\epsilon}}_t + \eta\tilde{\beta}_t\boldsymbol{\epsilon}$
end for
$\boldsymbol{x}_0 \leftarrow (\boldsymbol{x}_1 - \sqrt{1-\bar{\alpha}_1}\boldsymbol{\epsilon}_{\theta^*}(\boldsymbol{x}_1))/\sqrt{\bar{\alpha}_1}$
return $\boldsymbol{x}_0$

Algorithm 7 DDS (PI MRI; VE)

Require: $\boldsymbol{s}_{\theta^*}, N, \{\sigma_t\}_{t=1}^N, \eta, \boldsymbol{A}, M$

$\boldsymbol{x}_N \sim \mathcal{N}(\mathbf{0}, \sigma_T^2 \boldsymbol{I})$
for $t = N : 2$ **do**
  $\hat{\boldsymbol{s}}_t \leftarrow \boldsymbol{s}_{\theta^*}(\boldsymbol{x}_t)$
  $\hat{\boldsymbol{x}}_t \leftarrow \boldsymbol{x}_t + \sigma_t^2 \boldsymbol{s}_{\theta^*}(\boldsymbol{x}_t)$
  $\hat{\boldsymbol{x}}_t' \leftarrow \mathrm{CG}(\boldsymbol{A}^*\boldsymbol{A}, \boldsymbol{A}^*\boldsymbol{y}, \hat{\boldsymbol{x}}_t, M)$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\boldsymbol{x}_{t-1} \leftarrow \hat{\boldsymbol{x}}_t' - \sigma_{t-1}\sigma_t\sqrt{1-\tilde{\beta}^2\eta^2}\hat{\boldsymbol{s}}_t + \sigma_{t-1}\boldsymbol{\epsilon}$
end for
$\boldsymbol{x}_0 \leftarrow \boldsymbol{x}_1 + \sigma_1^2 \boldsymbol{s}_{\theta^*}(\boldsymbol{x}_1)$
return $\boldsymbol{x}_0$

Improvement on VE [31] Keeping the pre-trained model intact from [31], we switch from the Score-MRI sampling to Algorithm 7, and report on the reconstruction results from uniform 1D $\times 4$ accelerated measurements in Tab. 4.3. Note that Score-MRI uses 2000 PC as the default setting, which amounts to 4000 NFE, reaching 33.25 PSNR. We see almost no degradation in quality down to 200 NFE, but the performance rapidly degrades as we move down to 100, and completely fails when we set the NFE ≤ 50. On the other hand, by switching to the proposed solver, we are able to achieve the reconstruction quality that *better than* Score-MRI (4000 NFE) with only 100 NFE sampling. Moreover, we see that we can reduce the NFE down to 30 and still achieve decent reconstructions. This is a useful property for a reconstruction algorithm, as we can trade off reconstruction quality with speed.

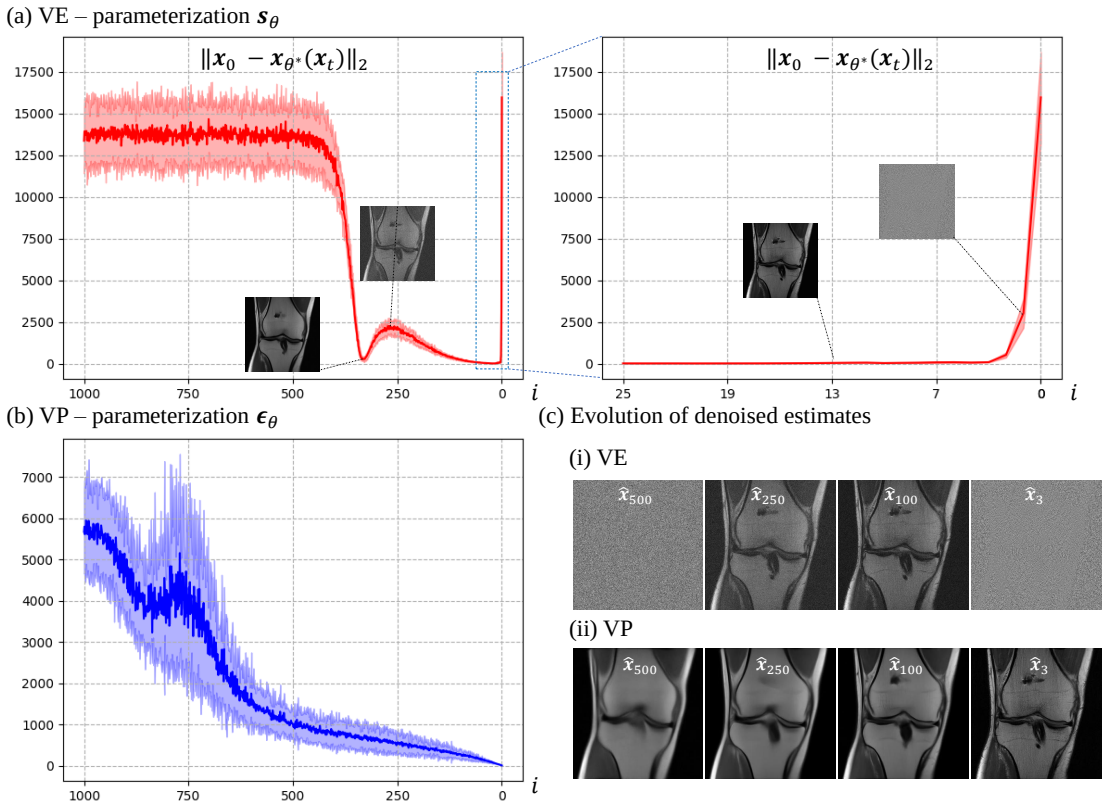


Figure 4.6: Evolution of the reconstruction error through time. $\pm 1.0\sigma$ plot. (a) VE parameterized with $\boldsymbol{s}_\theta$, (b) VP parameterized with $\boldsymbol{\epsilon}_\theta$, (c) Visualization of $\hat{\boldsymbol{x}}_t$.

NFE	4000	500	200	100	50	30
Score-MRI [31]	**33.25**	33.19	33.13	31.67	3.015	3.239
Ours	32.07	31.16	31.99	**33.69**	31.79	30.40

Table 4.3: PSNR [db] of uniform 1D $\times 4$ acc. reconstruction with varying NFEs.

One observation that is made in this experiment, however, is that using $\geq$ 200 NFEs for the proposed method *degrades* the performance. We find that this degradation stems from the numerical pathologies that arise when VE-SDE is combined with the parameterizing the neural network with $\boldsymbol{s}_\theta$. Specifically, the score function is parameterized to estimate $\boldsymbol{s}_{\theta^*}(\boldsymbol{x}_t) \simeq -\frac{\boldsymbol{x}_t - \boldsymbol{x}_0}{\sigma_t^2} = \boldsymbol{\epsilon}/\sigma_t$. Near $t = 0$, σ_t attains a very small value $e.g. \sigma_0 = 0.01$ [84], meaning that the score function has to approximate relatively large values in such regime, leading to numerical instability. This phenomenon is further illustrated in Fig. 4.6 (a), where the reconstruction (i.e. denoising) error has a rather odd trend of jumping up and down, and completely diverging as $t \to 0$. This may be less of an issue for using samplers such as PC where $\hat{\boldsymbol{x}}_i$ are not directly used but becomes a much bigger problem when the proposed sampler is used. In fact, for NFE $>$ 200, we find that simply truncating the last few evolutions is necessary to yield the result reported.

Such instabilities worsened when we tried scaling our experiments to complex-valued PI reconstruction due to the network only being trained on magnitude images. On the other hand, the reconstruction errors for VP trained with epsilon matching have a much stabler evolution of denoising reconstructions, suggesting that it is indeed a better fit in the context of the proposed methodology. Hence, all experiments reported hereafter use a network parameterized with $\boldsymbol{\epsilon}_\theta$ trained within a VP framework, and also by stacking real/imag part in the channel dimension to account for the complex-valued nature of the MR imagery and to avoid using $\times 2$ NFE for a single level of denoising.

Parallel Imaging with VP parameterization (Noiseless) We conduct thorough PI reconstruction experiments with 4 different types of sub-sampling patterns following [31]. See Fig. 4.7 for qualitative results. As the proposed method is based on diffusion models, it is agnostic to the sub-sampling patterns, generalizing well to all the different sampling patterns, whereas supervised learning-based methods such as U-Net and E2E-Varnet fail dramatically on 2D subsampling patterns.

We see that DDS sets the new state-of-the-art in most cases even when the NFE is constrained to < 100. Note that this is a dramatic improvement over the previous method [31], as for parallel imaging, Score-MRI required $120k(C = 15)$ NFE for the reconstruction of a single image. Contrarily, DDS is able to *outperform* score-MRI with 49 NFE, and performs *on par* with score-MRI with 19 NFE. Even disregarding the additional $\times 2C$ more NFEs required for score-MRI to account for the multi-coil complex valued acquisition, the proposed method still achieves $\times 80 \sim \times 200$ acceleration. We note that on average, our method takes about 4.7 seconds for 49 NFE, and about 2.25 seconds for 19 NFE on a single commodity GPU (RTX 3090).

Noisy multi-coil MRI reconstruction One of the most intriguing properties of the proposed DDS is the ease of handling measurement noise without careful computation of singular value decomposition (SVD), which is non-trivial to perform for our tasks. With Eq. (4.20), we can solve it with CG, arriving at Algo-

Mask Pattern	Acc.		TV	DPS (1000)	DDS VP (49)
Uniform 1D	$\times$ 4	PSNR [db]	24.19	24.40	**29.47**
		SSIM	0.687	0.656	**0.866**
	$\times$ 8	PSNR [db]	23.02	24.60	**26.77**
		SSIM	0.638	0.666	**0.827**
VD Poisson disk	$\times$ 8	PSNR [db]	23.07	23.48	**30.95**
		SSIM	0.609	0.592	**0.890**
	$\times$ 15	PSNR [db]	20.92	23.57	**29.36**
		SSIM	0.554	0.622	**0.853**

Table 4.4: Quantitative metrics for **noisy** parallel imaging reconstruction. Numbers in parenthesis: NFE.

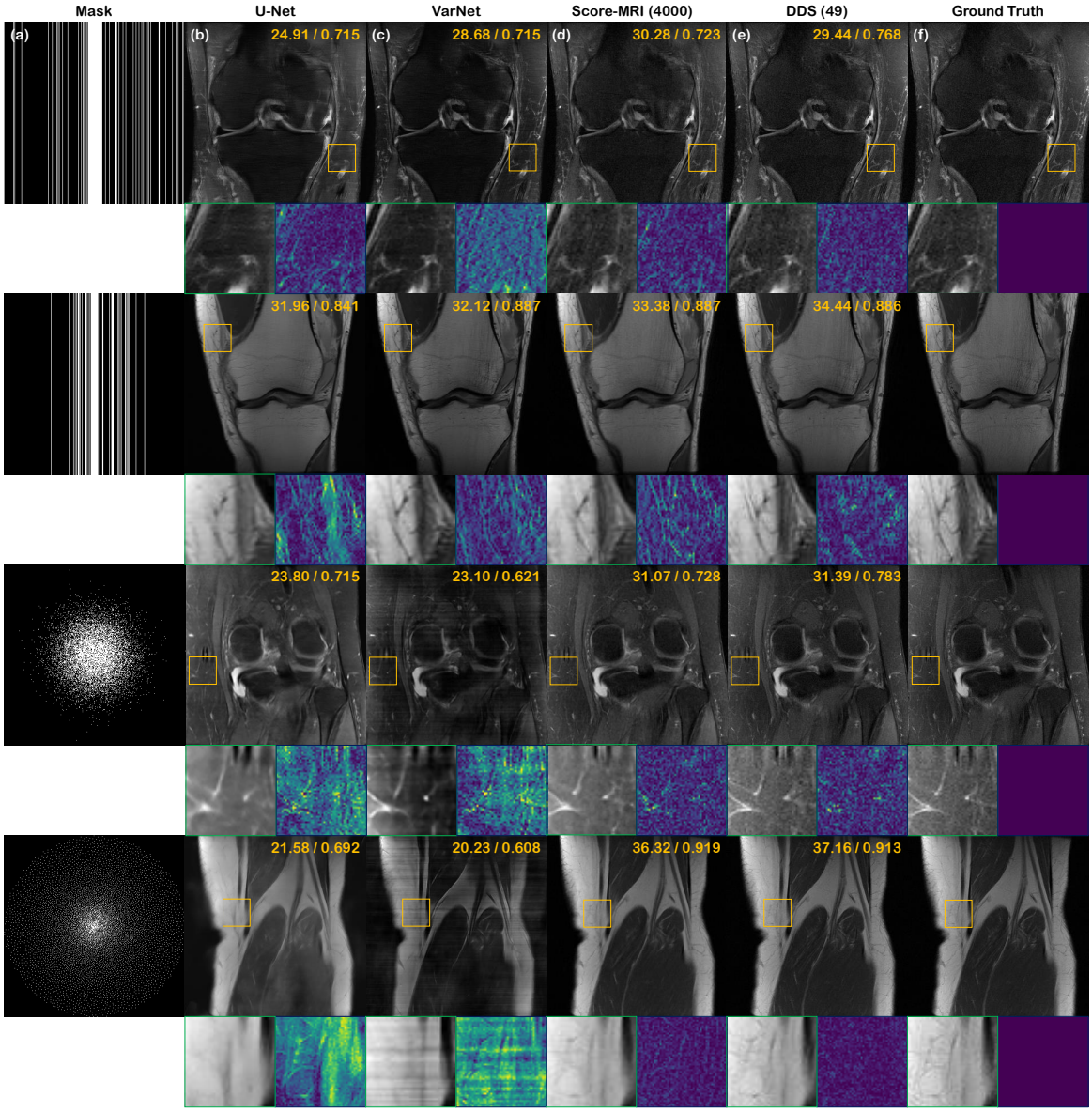


Figure 4.7: Comparison of parallel imaging reconstruction results. (a) subsampling mask (1st row: uniform1D$\times 4$, 2nd row: Gaussian1D$\times 8$, 3rd row: Gaussian2D$\times 8$, 4th row: variable density poisson disc $\times 8$), (b) U-Net [177], (c) E2E-VarNet [157], (d) Score-MRI [31] ($4000 \times 2 \times c$ NFE), (e) DDS (49 NFE), (f) ground truth.

rithm 6 in supplementary material. For comparison, methods that try to cope with measurement noise via SVD in the diffusion model context [82, 171] are not applicable and cannot be compared. One work that does not require computation of SVD and hence is applicable is DPS [26] relying on backpropagation. To test the efficacy of DDS on noisy inverse problems, we add a rather heavy complex Gaussian noise ($\sigma = 0.05$) to the k-space multi-coil measurements and reconstruct with Algorithm 6 by setting $\gamma = 0.95$ found through grid search. In Tab. 4.4, we see that DDS far outperforms DPS [26] with 1000 NFE by a large margin, while being about $\times 40$ faster as DPS requires the heavy computation of backpropagation.

4.3 Conclusion

In this chapter, we aimed for mitigating the downsides of the methods that were presented in chapter 3. We presented DiffusionMBIR, a method for solving 3D inverse problems in medical imaging by the use of factored priors, combining the strengths of diffusion model-based generative priors and simple model-based priors. Further, we presented DDS, a method to dramatically accelerate the inference of diffusion model-based inverse problem solvers to be applicable to large-scale inverse problems in medical imaging, including 3D. By studying the geometric properties of diffusion model, we showed that incorporating Krylov subspace methods could induce substantial gains in stability, performance, and speed.

Chapter 5. Deep Diffusion Image Prior

So far, we considered problems where the assumption was that we could accurately model the prior distribution $p_\theta(\boldsymbol{x})$ with a diffusion model. This is possible when we have access to high-quality "gold standard" data that are in-distribution to the test data. Unfortunately, there are numerous cases where the collection of high-quality gold standard data is not possible. For instance, it is difficult and expensive to collect large-scale data in medical imaging [177, 46], black hole imaging [33] and cryo-EM imaging [181, 56], etc. Consequently, one either has to resort to phantoms [47, 33] for generative modeling, or leverage implicit priors [181]. In these challenging cases, one faces an out-of-distribution (OOD) problem arising from mismatched priors [133], as the training data will be sufficiently different from the underlying true data distribution.

While it has been shown that DM-based inverse problem solvers (DIS) are less prone to distribution shifts [70, 31], the performance is significantly compromised [133, 5], leading to large gaps in performance. In the reconstructed images, the discrepancy in the distributions is characterized as artifacts and hallucinations. As there exist provable bounds in the performance [133] when considering standard DIS with fixed parameters, the goal is to *adapt* the parameters of the diffusion model so that it better covers the true distribution, even when all we have access to is a degraded measurement.

When the training data is unavailable, one of the most standard approaches in inverse imaging is the use of deep image prior (DIP) [162]

$$\theta^* = \arg\min_\theta \|\boldsymbol{y} - \boldsymbol{A}G_\theta(\boldsymbol{z})\|_2^2, \quad \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}), \tag{5.1}$$

where G_θ produces the reconstruction, and some means of early stopping is used to prevent the network from too much overfitting. As neural networks favor output signals that lie in the natural data manifold [162], and it can be considered a natural signal representation analogous to Fourier or Wavelets [66], optimizing for Eq. (5.1) leads to a decent reconstruction without any direct ground truth supervision. Over the years, there have been numerous advances in each of these components: design of the loss function, early stopping criterion, network parametrization, and initialization [75, 59, 3, 107, 6]. Despite the advances, DIP is still hard to optimize, and computationally demanding. For instance, applying DIP to a relatively small 3D data (167^3 resolution) requires about a day of training on a single RTX 3090 GPU [6].

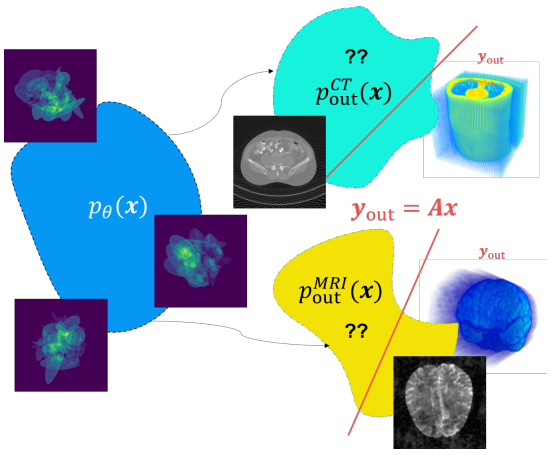


Figure 5.1: OOD inverse problem setting. Pre-trained diffusion model learns $p_\theta(\boldsymbol{x})$, but at test time we only have $\boldsymbol{y}_{\text{out}}$ obtained from unknown OOD distributions, and aim to sample from $p_{\text{out}}(\boldsymbol{x}|\boldsymbol{y}_{\text{out}})$.

In this section, we extend the idea of DIP into the realm of diffusion inverse solvers, showing that we can yield a superior algorithm in speed, robustness, and quality by leveraging the merits of diffusion models. Two works that are perhaps the most related are Educated DIP [6] (EDIP) and Baguer *et al.* [3]. EDIP [6] shows that initializing G_θ with a network trained for reconstruction helps in convergence. [3] proposes to use some initial reconstruction as an input to G_θ, rather than some random vector $\boldsymbol{z}$. As will be later seen, our method naturally leverages both of

these techniques by pivoting along the PF-ODE path.

5.1 DDIP: OOD adaptation in diffusion inverse solvers

For simplicity in exposition, let us define the simplest diffusion PF-ODE, constructed from $p(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_0, t^2\boldsymbol{I})$

$$d\boldsymbol{x}_t = -t\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)\, dt = \frac{\boldsymbol{x}_t - \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]}{t}\, dt, \quad p(\boldsymbol{x}_T) \sim \mathcal{N}, \tag{5.2}$$

When aiming for *posterior* sampling, one can additionally condition the PF-ODE with $\boldsymbol{y}$, which reads

$$d\boldsymbol{x}_t = -t\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{y})\, dt = \frac{\boldsymbol{x}_t - \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}]}{t}\, dt, \quad p(\boldsymbol{x}_T) \sim \mathcal{N}. \tag{5.3}$$

As $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}]$ (equivalently $\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{y})$) is intractable, various methods have been proposed to approximate the posterior sampling process [26, 171, 183, 27]. DPS approximates

$$\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}] = \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t] + t^2\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{y}|\boldsymbol{x}_t) \approx \hat{\boldsymbol{x}}_{0|t} + \frac{t^2}{2\sigma_y^2}\nabla_{\boldsymbol{x}_t}\|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_{0|t}\|_2^2, \tag{5.4}$$

with $\hat{\boldsymbol{x}}_{0|t} := D_{\theta^*}(\boldsymbol{x}_t)$[1]. As taking backprop w.r.t. $\boldsymbol{x}_t$ through the network is expensive and unstable [129], a way to avoid this was proposed in DDS, which uses

$$\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}] \approx \arg\min_{\boldsymbol{x}_0} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_0\|_2^2 + \frac{\gamma}{2}\|\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_{0|t}\|_2^2, \tag{5.5}$$

solved with M-step conjugate gradient (CG), which we simply denote as $\texttt{CG}(\hat{\boldsymbol{x}}_{0|t}, M)$. Existing DIS can be considered as different ways of approximating the *empirical* conditional posterior, which we denote as $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}] \approx D_\theta(\boldsymbol{x}_t|\boldsymbol{y})$. Unfortunately, under mismatched prior, the approximation of the conditional posterior mean given by $D_\theta(\boldsymbol{x}_t|\boldsymbol{y})$ will be highly inexact, containing hallucinations from the training distribution which may not be desirable. We explore ways in which one can mitigate these by adapting the parameters of the diffusion model in the following section.

Recall that DIP optimizes the network parameters with the fidelity loss in Eq. (5.1). As the optimization is held specific to $\boldsymbol{y}$, Eq. (5.1) aims to implicitly recover the posterior mean $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{z}, \boldsymbol{y}]$. On the other hand, DIS at time t during sampling, produces some estimate of the conditional posterior mean $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}] \approx D_\theta(\boldsymbol{x}_t|\boldsymbol{y})$. As $\boldsymbol{x}_t$ is equivalent to $\boldsymbol{z}$ at $t = T$ (initial noise), we can generalize DIP in Eq. (5.1) to multi-scale DIP over multiple noise scales

$$\text{for } t = T, \ldots, 1 : \theta_{t-1} \leftarrow \arg\min_{\theta_t} \|\boldsymbol{y} - \boldsymbol{A}D_{\theta_t}(\boldsymbol{x}_t|\boldsymbol{y})\|_2^2, \tag{5.6}$$

$$\boldsymbol{x}_{t-1} \leftarrow \text{DDIM}_{\theta_{t-1}}(D_{\theta_{t-1}}(\boldsymbol{x}_t|\boldsymbol{y})) \tag{5.7}$$

where Eq. (5.7) denotes some diffusion inverse problem solver that proceeds by leveraging the adapted parameters θ_{t-1}. It is easy to see that Eq. (5.6) is equivalent to Eq. (5.1) when $t = T$, but it is beneficial because 1) the optimization steps are performed in a coarse-to-fine manner starting from large noise scale, 2) the trajectory is pivoted along the original PF-ODE trajectory, providing a good initialization. Our approach introduces a better initialization in the network parameters as in [6], and also a better initialization in the input to the network [3] for

[1]This notation is taken for simplicity. When emphasizing the dependence, we denote $\hat{\boldsymbol{x}}_{0|t}(\boldsymbol{x}_t; \theta^*)$

Algorithm 8 DDIP

Require: $\theta, D^t, N, T', \eta, \boldsymbol{y}$
$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, D_\theta) := \|\boldsymbol{y} - \boldsymbol{A}D_\theta(\boldsymbol{x}|\boldsymbol{y})\|_2^2$
for $i = 1$ **to** N **do**
  $\theta_{T'}^{(i)} \leftarrow \theta$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$
  $\boldsymbol{x}_T^{(i)} \leftarrow \sqrt{\bar{\alpha}_{T'}}\boldsymbol{A}^\dagger \boldsymbol{y} + \sqrt{1 - \bar{\alpha}_{T'}}\boldsymbol{\epsilon}$
  for $t = T'$ **to** 1 **do**
    $\theta_{t-1}^i \leftarrow \arg\min_{\theta_t^i} \mathcal{L}(\boldsymbol{x}_t^i, \boldsymbol{y}^i, D_{\theta_t^i}^t)$
    $\hat{\boldsymbol{x}}_{0|t}^i \leftarrow D_{\theta_{t-1}^i}^t(\boldsymbol{x}_t^i|\boldsymbol{y}^i)$
    $\boldsymbol{x}_{t-1}^i \leftarrow \mathrm{DDIM}(\hat{\boldsymbol{x}}_{0|t}^i, \eta)$
  end for
  return $\boldsymbol{x}_0^i$
end for

Algorithm 9 D3IP

Require: $\theta, D^t, N, T', K, \eta, \boldsymbol{Y}$
$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, D_\theta) := \|\boldsymbol{y} - \boldsymbol{A}D_\theta(\boldsymbol{x}|\boldsymbol{y})\|_2^2$
$\theta_{T'} \leftarrow \theta$
$\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_N \sim \mathcal{N}(0, \boldsymbol{I})$
$\boldsymbol{\epsilon} \leftarrow \mathrm{slerp}(\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_N, \frac{i}{N})$
$\boldsymbol{X}_{T'} \leftarrow \sqrt{\bar{\alpha}_{T'}}\boldsymbol{A}^\dagger \boldsymbol{Y} + \sqrt{1 - \bar{\alpha}_{T'}}\boldsymbol{\epsilon}$
for $t = T'$ **to** 1 **do**
  $\boldsymbol{x}_t^{\{i\}}, \boldsymbol{y}^{\{i\}} \sim \mathrm{MC}((\boldsymbol{X}_t, \boldsymbol{Y}), K)$
  $\theta_{t-1} \leftarrow \arg\min_{\theta_t} \mathcal{L}(\boldsymbol{x}_t^{\{i\}}, \boldsymbol{y}^{\{i\}}, D_{\theta_t}^t)$
  $\hat{\boldsymbol{X}}_{0|t} \leftarrow D_{\theta_{t-1}}^t(\boldsymbol{X}_t|\boldsymbol{Y})$
  $\boldsymbol{X}_{t-1} \leftarrow \mathrm{DDIM}(\hat{\boldsymbol{X}}_{0|t}, \eta)$
end for
return $\boldsymbol{X}_0$

$t < T$. We define a general method presented in Eq. (5.6), Eq. (5.7) as deep diffusion image prior (DDIP). To keep the power of the original network, in practice, we introduce LoRA [67] parameters, and only update the newly introduced parameters.

5.1.1 Extending DDIP to 3D

Let $\boldsymbol{X} \in \mathbb{R}^{n \times N} = [\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}]$ a stacked 3D tensor with N slices, and $\boldsymbol{Y}$ its corresponding measurement. In [5], it was proposed to independently run adaptation for every different slice, which requires a prohibitively long amount of time for adaptation/sampling. Concretely, with a 256^3-sized volume that we consider in this work, it requires 90 seconds per slice, which is > 6 hours per volume.

For brevity, let us view Eq. (5.6) as minimizing an expected value $\mathbb{E}_t\left[\|\boldsymbol{y} - \boldsymbol{A}D_\theta(\boldsymbol{x}_t|\boldsymbol{y})\|_2^2\right]$ for every slice i. In SCD, the authors solve this optimization independently across all slices, as illustrated in Fig. 5.2 (a). Instead, we propose to optimize Eq. (5.6) so that the adaptation can be done synergistically in expectation (See Fig. 5.2 (b) for an illustration)

$$\min_\theta \mathbb{E}_i\left[\mathbb{E}_t\left[\|\boldsymbol{y}^i - \boldsymbol{A}D_\theta^t(\boldsymbol{x}_t^i|\boldsymbol{y}^i)\|_2^2\right]\right] \approx \frac{1}{K}\sum_{i=1}^K \mathbb{E}_t\left[\|\boldsymbol{y}^i - \boldsymbol{A}D_\theta^t(\boldsymbol{x}_t^i|\boldsymbol{y}^i)\|_2^2\right], \tag{5.8}$$

where i denotes the index across the slices, and we can use K Monte Carlo samples to compute the expectation. In practice, we find even using a small K suffices for stable optimization and the performance plateaus for $K > 6$, yielding a compute-effective algorithm. The formulation of Eq. (5.8) lets us adapt the parameters that are suited for the whole 3D volume so that the memory and computation are reduced to $\mathcal{O}(1)$. We name our method D3IP (base), which will be shown to be improvable in specific settings, as we investigate in the following sections. We highlight the difference of D3IP against DDIP in Alg. 8,9, where $\mathrm{MC}(\cdot, K)$ in L6 of Alg. 9 represents K-Monte Carlo sampling, and $\boldsymbol{x}^{\{i\}}$ denotes the sampled vectors stacked into a single tensor. Surprisingly, not only is D3IP an order of magnitude cheaper and faster than DDIP, but it performs *better* than DDIP. This can be attributed to D3IP learning from a broader and richer context, whereas DDIP is only allowed to learn from a single slice of information.

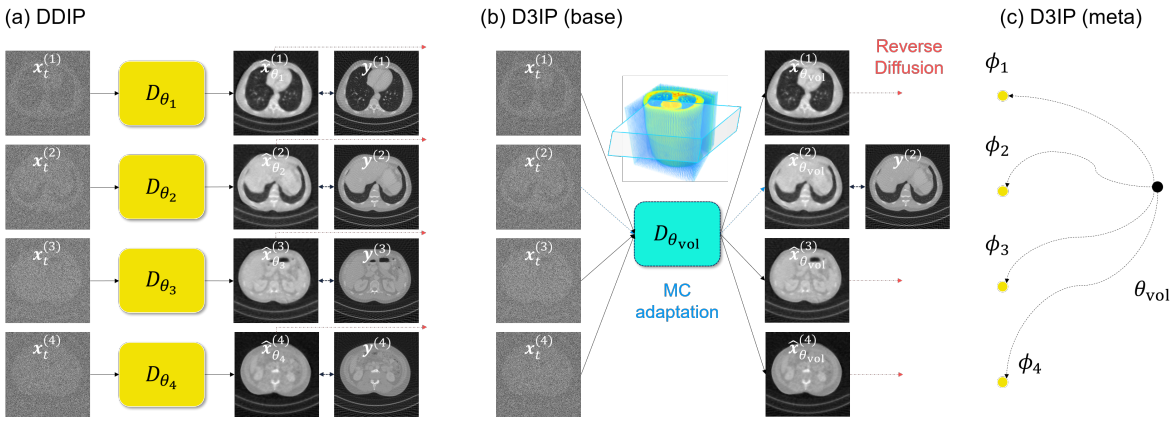


Figure 5.2: OOD adaptation schemes in DIS. (a) DDIP/SCD performs *independent* adaptation across slices and requires $\mathcal{O}(N)$ compute & memory. (b) D3IP (base) performs joint adaptation with stochastic gradients from MC sampling (blue dotted line) and requires $\mathcal{O}(1)$ compute & memory. (c) θ_{vol} adapted from D3IP (base) can be used as a meta-parameter to be further adapted to specific slices.

5.1.2 Incorporating 3D DIS to D3IP

Another big advantage of D3IP is that we can now seamlessly integrate 3D DIS methods [28, 100] into our framework. As [100] would require adapting two diffusion models operating on different orientations, we choose DiffusionMBIR[28] accelerated with DDS[27] as our approximator (simply denoted DiffusionMBIR hereafter). Concretely, the approximation reads

$$D_\theta^t(\boldsymbol{X}_t|\boldsymbol{Y}) = \arg\min_{\boldsymbol{X}_0} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{X}_0\|_2^2 + \frac{\gamma}{2}\|\boldsymbol{T}\boldsymbol{X}_0\|_1, \tag{5.9}$$

where $\boldsymbol{T}$ computes the finite difference along the stacked slice dimension, and the optimization is solved with a few-step ADMM [11] initialized with $\hat{\boldsymbol{X}}_{0|t}(\boldsymbol{X}_t;\theta_t)$ to induce proximal regularization.

One caveat is that Eq. (5.9) requires computing total variation (TV) regularization for adjacent slices, which cannot be done if we randomly sample K slices as done in D3IP (base). Instead, this can be effectively approximated by sampling $K < N$ neighboring slices, i.e. modifying L6 of Alg. 9 to

$$\boldsymbol{x}_t^{i:i+K-1}, \boldsymbol{y}^{i:i+K-1} \sim \text{MC}((\boldsymbol{X}_t, \boldsymbol{Y}), K). \tag{5.10}$$

We refer to this method with D3IP (mbir). The regularization in Eq. (5.9) is already useful for the OOD setting as it is independent of training data, and 3D samplers typically provide a better estimate in 3D inverse problem settings. Consequently, we observe improved performance in the adapted setting by simply incorporating a 3D DIS.

5.1.3 Meta-learning D3IP

The main motivation of D3IP was to reduce the memory and computation cost for OOD adaptation of diffusion models. Nevertheless, we see in practice that D3IP does not always achieve better performance than SCD, which can be attributed to the fact that the parameters are being adapted to an *average* direction. Negative transfer classically arising in multi-task optimization [34, 38] can lead to a suboptimal result for each slice.

On the other hand, consider meta-learning [118, 51], where the objective is to find a good *meta* parameter that can be quickly adapted to different tasks that one considers. Interestingly, our formulation can be cast as a meta-learning problem when we view the optimization problem with respect to each slice, a single task. Recall that

the Reptile algorithm [118], at the t-th step going on to $t-1$-th step, follows

$$\tilde{\theta}_t \leftarrow U^c_{\{i\}}(\theta_t), \quad \theta_{t-1} \leftarrow \theta_t + \alpha(\tilde{\theta}_t - \theta_t), \quad \alpha_t \in (0, 1], \tag{5.11}$$

where $U^c_{\{i\}}(\cdot)$ is c-step gradient update using an optimizer where the sampled tasks are in $\{i\}$. Interestingly, under this view, Alg. 9 is already the Reptile algorithm [118] where the optimization problem of L7 is solved with $U^c_{\{i\}}(\cdot)$, and a constant step size $\alpha_t = 1.0$ is chosen, corresponding to making full updates. Following [118], we define D3IP (meta) by choosing a value of α_t to linearly decay starting from 1.0 at the initial iteration, providing a better initialization point to be further fitted to each slice. We refer to the meta-parameter as θ_{vol}.

After running the meta-learning algorithm, when one is willing to trade-off more compute with better performance, we can further fine-tune our adapted meta-parameter θ on respective slices to obtain a parameter set $\{\phi_1, \dots, \phi_N\}$ with the usual DDIP, initializing Alg. 8 for every slice from θ_{vol}, achieving higher quality reconstruction than standard DDIP without meta-learning. The illustration of the idea is shown in Fig. 5.2 (c).

5.1.4 Technical Advances

On top of the fundamental innovations, we propose several technical advances to the adaptation algorithm that yield faster and more robust optimization. These advances are orthogonal to the contributions that are proposed in the rest of the section, and can be applied to all adaptation methods.

Constraining the optimization horizon. It is known that the empirical score functions exhibit problematic behavior in the end regimes [79, 84] as the estimation tends to be inaccurate and volatile. Motivated from the score distillation sampling literature [129, 172], we truncate the regime where we optimize for Eq. (5.6) or Eq. (5.8), so that for $t \notin [\zeta, T-\zeta]$, we only run standard DIS, with $\zeta = 40$ unless specified otherwise. We find that including adaptation outside this regime deteriorates the performance.

Initialization strategy. It is standard practice to initialize DIS with random Gaussian noise [82, 26, 152, 171]. However, it is also known that due to the non-zero terminal SNR of diffusion models [103], the low-frequency component of the initialization is carried out to the sample. Due to this property, some works propose to initialize the Gaussian noise with low-frequency part replaced from a similar sample [174]. In the case of inverse problems, this often comes for free. For instance, one can use the pseudo-inverse of the measurement. Moreover, motivated by the initialization strategy for DIP [175] and diffusion models [54] for video, which has correlations across time frames, we propose to sample two random noise vectors for the end slices and leverage the spherically interpolated (slerp) vector for the initialization noise. Summing up, our initialization reads

$$\boldsymbol{x}^{(i)}_{T'} \leftarrow \sqrt{\bar{\alpha}_{T'}}\boldsymbol{A}^\dagger \boldsymbol{y}^{(i)} + \sqrt{1-\bar{\alpha}_{T'}}\text{slerp}(\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_N, \frac{i}{N}), \quad \boldsymbol{\epsilon}_0, \boldsymbol{\epsilon}_N \sim \mathcal{N}(0, \boldsymbol{I}), \tag{5.12}$$

with $T' < T$ typically set to 980. This not only lets us start from a correlated initial noise vector with low-frequency components consistent with the measurement but also lets us avoid any instabilities arising from the non-zero terminal SNR problem [103].

5.1.5 Experiments

We consider three canonical inverse problems in medical imaging: 3D sparse-view CT reconstruction (3D SV-CT), 3D MRI reconstruction (3D MRI), and multi-coil MRI reconstruction (CS-MRI), as these are the cases where the different measurement slices originate from the same volume. For the CT reconstruction task, we have a few hundred test slices originating from the same volume. For the first two tasks, we use a diffusion prior trained only on ELLIPSES phantoms [1] generated on-the-fly, which are completely irrelevant to the target data distribution.

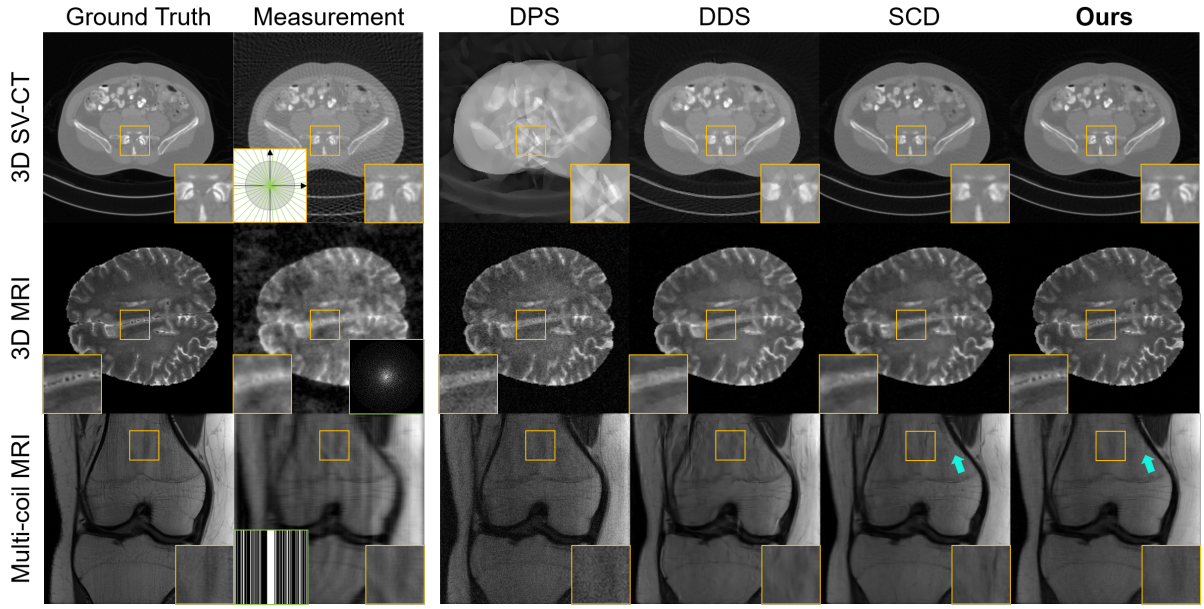


Figure 5.3: Representative results on 3 different tasks. (row 1-2): 3D SV-CT, (row 3-4): 3D MRI, (row 5-6): CS-MRI. Comparison against DPS [26], DDS [27], and SCD [5]. **Ours**: D3IP (base). Cyan arrows indicate regions of remaining artifacts even after adaptation with SCD. Green boxes illustrate the acquisition scheme of the measurement (acquisition angle, sub-sampling pattern).

We focus on such setting this leads to a completely unsupervised approach for a reconstructive task, requiring no collection of gold standard data.

For the MRI reconstruction task, we consider two different realistic cases: when the acquisition scheme is truly 3D and the volume consists of a few hundred slices, and when the acquisition scheme is 2D but the slices originate from the same volume, yielding a few ten slices. For all tasks, we consider variance preserving (VP) diffusion models trained under the ADM [39] framework unless specified otherwise.

Experimental settings

3D SV-CT. Following [5], we take a diffusion model trained on the ELLIPSES dataset [1] and test it on a volume of the American Association of Physicists in Medicine (AAPM) grand challenge [115] dataset, following the settings of [29, 28]. This is a particularly useful and interesting setting, as the ELLIPSES dataset are phantoms that can be easily generated on-the-fly, requiring *no* collection of data, a realistic condition when acquiring high-quality ground truth images is impossible. We consider parallel CT geometry with 60-angle measurements to be consistent with [5].

3D MRI reconstruction We take the same ELLIPSES diffusion model used for 3D SV-CT, and adapt it to the multimodal brain tumor image segmentation benchmark (BRATS) 2018 [114]. The test set consists of 5 T1 contrast / 5 T2 contrast volumes. We consider variable density (VD) Poisson disc sampling pattern [42] ($\times 8$ acceleration) for 3D volume measurements in a single-coil setting.

2D Multi-coil MRI reconstruction. A diffusion model trained on fastMRI [177] BRAIN data was taken from [27][2]. The evaluation was done on fastMRI KNEE data, consisting of 10 test volumes and 294 slices. The measurement was simulated using the uniform 1D subsampling ($\times 4$ acceleration) with 8% Auto Calibrating Signal (ACS) region, as proposed in [177]. Since we consider multi-coil measurements in this task, the coil sensitivity maps are estimated using ESPiRiT [161].

[2] https://github.com/HJ-harry/DDS

Method	3D SV-CT (ELLIPSES → AAPM)			3D MRI (ELLIPSES → BRATS)			Mult. coil CS-MRI (BRAIN → KNEE)			Compute	Memory
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	$^*K \ll T < N$	
DDNM [171]	23.55	0.765	0.241	14.59	0.281	0.753	19.35	0.354	0.493	$\mathcal{O}(T)$	$\mathcal{O}(1)$
DPS [26]	17.85	0.470	0.463	27.30	0.394	0.410	27.26	0.732	0.312	$\mathcal{O}(T)$	$\mathcal{O}(1)$
DDS [27]	27.65	0.805	0.188	24.59	0.532	0.339	28.36	0.741	0.278	$\mathcal{O}(T)$	$\mathcal{O}(1)$
DiffusionMBIR [28]	28.92	0.845	0.199	26.97	0.620	0.299	-	-	-	$\mathcal{O}(T)$	$\mathcal{O}(1)$
SCD [5]	32.91	0.904	0.184	26.00	0.561	0.338	29.01	0.752	0.269	$\mathcal{O}(KNT)$	$\mathcal{O}(N)$
DDIP	33.13	0.903	0.164	27.46	0.647	0.289	29.11	0.775	0.246	$\mathcal{O}(KNT)$	$\mathcal{O}(N)$
D3IP (base)	31.73	0.908	0.141	30.59	0.859	0.152	29.00	**0.789**	0.233	$\mathcal{O}(KT)$	$\mathcal{O}(1)$
D3IP (mbir)	33.69	**0.919**	**0.133**	**33.89**	**0.907**	**0.103**	-	-	-	$\mathcal{O}(KT)$	$\mathcal{O}(1)$
D3IP (meta)	**33.96**	0.917	0.136	32.60	0.877	0.126	**29.52**	0.779	**0.216**	$\mathcal{O}(KNT)$	$\mathcal{O}(N)$

Table 5.1: Quantitative measure of OOD Inverse problem solving on 3 main tasks.

For all inverse problems considered, we add $\sigma_y = 0.01$ measurement noise when simulating with forward operators.

Diffusion samplers

For all DIS methods including the proposed method, we consider 50 NFE DDIM sampling with $\eta = 0.85$ unless specified otherwise. One exception is DPS [26], where we take 1000 NFE with $\eta = 1.0$ (i.e. DDPM sampling) to achieve satisfactory results.

Adaptation. For the family of the proposed method (DDIP, D3IP), we take DDS [27] as our estimator D_θ for $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}]$, which runs 5 CG optimization step per diffusion denoising step. In 3D SV-CT and 3D MRI tasks, K is set to 6, and for CS-MRI, $K = 3$. For D3IP (mbir), we use the approximation in Eq. (5.9), which is solved with 5 ADMM iterations per diffusion step. D3IP (meta) runs D3IP in Alg. 9 with linearly decreasing step size from $\alpha = 1.0$ to $\alpha = 0.5$ in Eq. (5.11). The obtained meta-parameter θ_{vol} is then fine-tuned with respect to each slice using DDIP in Alg. 8 initialized from this meta-parameter.

Comparison methods. We consider several strong DIS baselines: DDNM [171], DPS [26], DDS [27], DiffusionMBIR [28], and SCD [5] for comparison. In order to apply DDNM, one needs access to the pseudo-inverse operator, which is often hard to compute, or numerically unstable. We circumvent this by leveraging CG updates motivated from [27]. We note that DiffusionMBIR is the only 3D-aware DIS, and SCD is the only adaptation-based DIS among the comparison methods.

Method	PSNR ↑	SSIM ↑
DDIP(baseline)	35.62	0.908
+ constrained horizon	35.98	0.916
+ init. strategy	**36.31**	**0.922**

Table 5.2: Improvements from configurations introduced in Sec. 5.1.4.

Results

Ablation study on the technical advances. Before diving into the main results, we first conduct an ablation study introduced in Sec. 5.1.4, as these techniques can be leveraged by all adaptation samplers. In Tab. 5.2, it is evident that the techniques provide improvements in the metrics, hence we keep the final configuration through all our experiments.

Main Results. In Tab. 5.1, we present a thorough comparison study on all three tasks that we consider in this work. Here, we see that D3IP not only dramatically reduces the computation cost of the adaptation, but also often results in a *better* reconstruction quality than DDIP. Specifically, this effect is most evidently seen in the 3D MRI task, where there is more than 3 dB difference in PSNR, and the perceptual LPIPS metric improves also by a

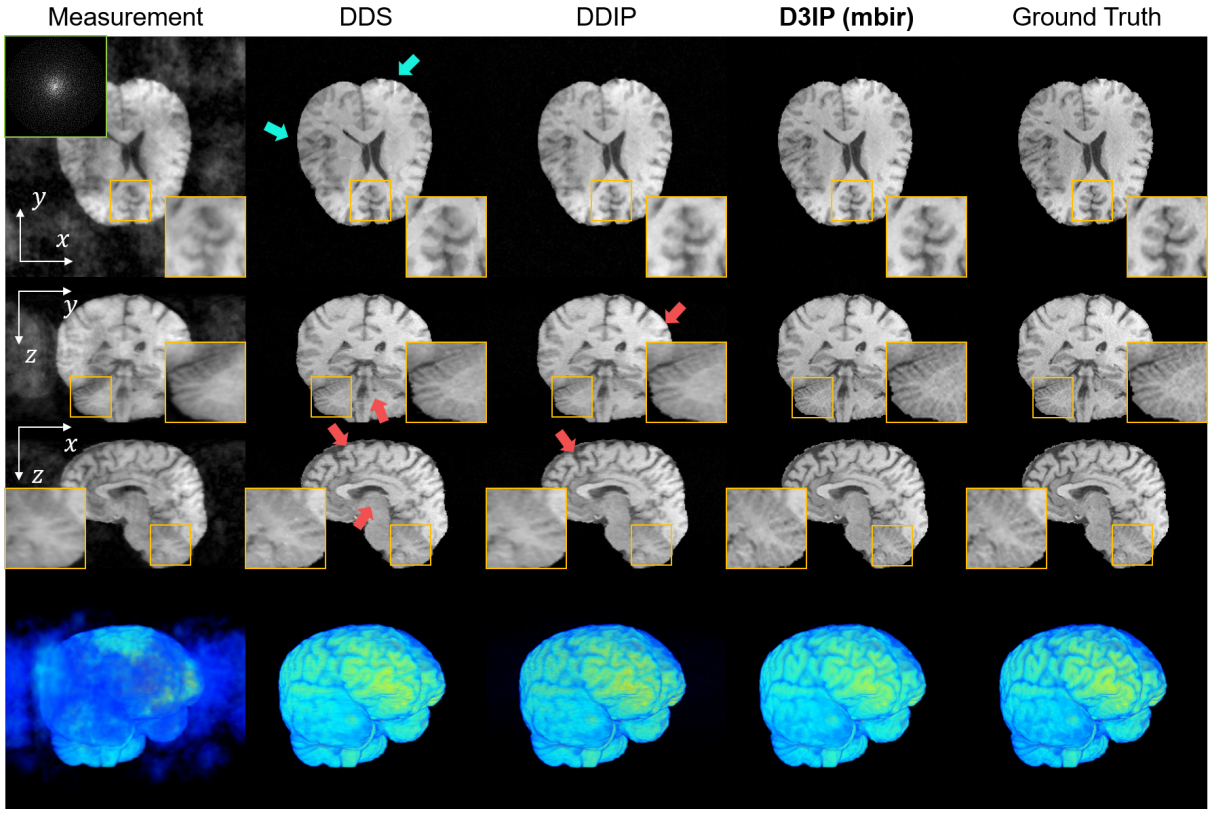


Figure 5.4: 3D-MRI reconstruction with DDS [27], DDIP, D3IP (mbir). Cyan and red arrows indicate artifacts from prior mismatch and slice-wise independent reconstruction, respectively. 1-4$^{\text{th}}$ row: xy, yz, xz slice, and 3D rendering.

large margin. Among all the tasks and the metrics considered, only the PSNR values of 3D SV-CT and Multi-coil CS-MRI fall short of DDIP. For the rest of the cases, D3IP outperforms DDIP, thanks to the richer information provided during the optimization process.

In Fig. 5.3, we compare the qualitative results of our proposed D3IP (base) solver against SOTA DIS - DPS, DDS, and the previous adapted sampler SCD. It is clear that the results obtained with DPS and DDS are contaminated with artifacts and hallucinations that originate from the training data (elliptical phantoms). While SCD greatly alleviates these artifacts, the details are typically blurred out, erasing structures that can even be seen from the measurement (e.g. 1st row of Fig. 5.3). Notably, this happens even without any TV regularization that was used in [5]. Introducing such regularization yields even blurrier results. In contrast, the proposed method is capable of restoring crisp features without such blurring. Moreover, D3IP further eliminates leftover artifacts that are still apparent in SCD, as pointed out with cyan arrows.

It should be noted that all these advances take place while also being much faster and cheaper in memory. On a single 3090 RTX GPU, for a 256^3 volume, D3IP (base) requires 40 minutes, whereas SCD requires $\sim$6.2 hours, as it should be conducted slice-wise. Moreover, by using D3IP, we can reduce the memory requirements of the LoRA parameters from 2.9 Gb to 14.5 Mb as we only need to keep a single set of parameters for the whole volume, rather than specific parameters for each slice.

Incorporating DiffusionMBIR. While D3IP (base) enables the use of average gradients computed from multiple slices, it does not guarantee smooth reconstruction across slices, which can be effectively counteracted by leveraging 3D solvers. In Fig. 5.4, we visualize the 3D rendering as well as the coronal and the sagittal slices of the reconstruction, highlighting the advantage of the proposed method by allowing a plug-and-play incorporation of

3D DIS.

Meta-learning further improves D3IP. By initializing a meta-parameter θ_{vol} and further adapting the parameters toward specific slices, we see in Fig. 5.5 and Tab. 5.1 that we can achieve even better performance by trading off compute. In contrast, gradually adapting the parameters across different slices with SCD does not lead to noticeable improvements, compared to the case where the weights are re-initialized for every slice.

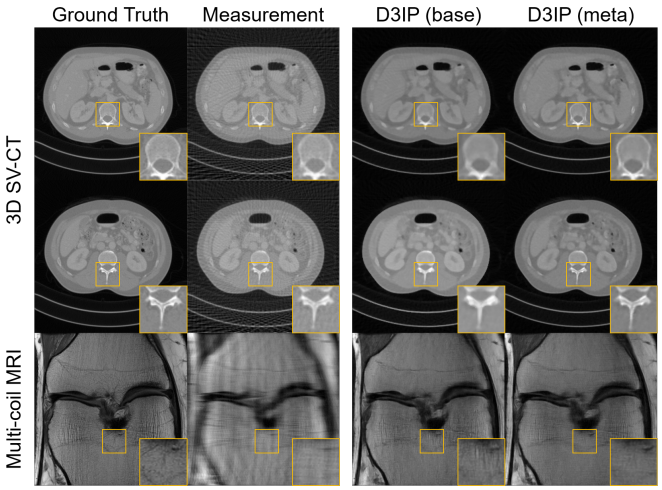


Figure 5.5: Comparison of reconstructions with D3IP (base) and D3IP (meta).

5.2 Conclusion

In this chapter, we proposed DDIP, a method of adapting diffusion priors trained on OOD distribution for solving inverse imaging tasks. We clarified that DDIP is a generalization of DIP constructed on the PF-ODE trajectory of diffusion, which makes the method much stabler than standard DIP. Focusing on 3D inverse problems, we proposed D3IP, which significantly reduces the computation cost while achieving better performance. We showed that D3IP can be further enhanced by incorporating 3D inverse problem solvers, or by leveraging meta-learning to induce a meta-parameter, then fine-tuning to specific 2D measurements. Notably, our work improves previous approaches in reconstruction quality, closing the gap against in-distribution DIS; it accelerates the speed of the algorithm to a regime where it will be practical for real-world usage; it enhances the interpretability of the algorithm and clarifies why the method works. We believe that our work can become the ground for many inverse imaging applications where unsupervised reconstruction remains crucial, such as in biomedical imaging and astronomy.

Chapter 6. Text-driven Inverse Problems

The classic meaning of inverse problem solving, including the methods treated up to this chapter, considers retrieving $\boldsymbol{x}$ solely from the information obtained in $\boldsymbol{y}$, i.e. the sensor information. However, information is rarely captured in a unimodal fashion. For instance, in MRI, alongside capturing the k-space signal, we have access to patient information, MR imaging parameters, and medical history. Unfortunately, this additional information is always discarded during inference—a suboptimal practice according to the data processing inequality.

Up until now, we did not have the right tools to practically incorporate metadata information into the recovery. However, the recent surge of text-to-image diffusion models [135, 141] indicate that this is no longer the case: it is possible to train a good text-conditional generative prior. In this chapter, we explore the possibility of leveraging texts as additional metadata to improve the performance of signal recovery, as well as use it to guide the solution to a specific mode.

6.1 P2L: prompt-tuning latent diffusion models for inverse problems

Solving inverse problems in a fully general domain is hard. This directly stems from the difficulty of generative modeling a wide distribution, where it is known that one has to trade-off diversity with fidelity by some means of sharpening the distribution [13, 39]. The standard approach in modern diffusion models is to condition on text prompts [135, 141], among them the most popular being Stable Diffusion (SD), a latent diffusion model (LDM), which is itself an under-explored topic in the context of inverse problem solving. While text conditioning is now considered standard practice in content creation including images [131, 141], 3D [129, 172], video [61], personalization [52], and editing [60], it has been completely disregarded in the inverse problem solving context. This is natural, as it is highly ambiguous which text would be beneficial to use when all we have is a degraded measurement. The wrong prompt could easily lead to degraded performance.

In this work, we aim to bridge this gap by proposing a way to *automatically* find the right prompt to condition diffusion models when solving inverse problems. This can be achieved through optimizing the continuous text embedding *on-the-fly* while running DIS. We formulate this into a Bayesian framework of updating the text embedding and the latent in an alternating fashion, such that they become gradually aligned during the sampling process. Orthogonal and complementary to embedding optimization, we devise a simple LDM-based DIS (LDIS) that controls the evolution of the latents to stay on the natural data manifold and additionally utilizes the VAE prior for stability of the solutions. We name the algorithm that combines these components P2L, short for **P**rompt-tuning **P**rojected **L**atent diffusion model-based inverse problem solver. In reaching for the ultimate goal of DIS, we focus on 1) **LDM-based DIS** (LDIS) for solving inverse problems in the 2) **fully general domain** (using a single pre-trained checkpoint) that targets 3) **512×512 resolution**[1]. All the aforementioned components are highly challenging, and to the best of our knowledge, have not been studied in conjunction before.

[1]All prior works on DIS/LDIS focused on 256×256 resolution. Most LDIS focused their evaluation on a constrained dataset such as FFHQ, and did not scale their method to more general domains such as ImageNet.

	FFHQ						ImageNet					
	SR×8			Inpaint ($p = 0.8$)			SR×8			Inpaint ($p = 0.8$)		
Prompt	FID↓	LPIPS↓	PSNR↑	FID↓	LPIPS↓	PSNR↑	FID↓	LPIPS↓	PSNR↑	FID↓	LPIPS↓	PSNR↑
`""`	61.16	0.327	26.49	52.34	0.241	**29.78**	78.68	0.397	23.49	70.87	0.350	26.20
`"A high quality photo"`	61.17	0.327	26.57	52.82	0.237	29.70	77.00	0.396	23.51	69.10	0.350	26.26
`"A high quality photo of a cat"`	69.03	0.377	26.39	55.15	0.248	29.63	76.69	0.402	**23.63**	68.48	0.355	26.13
`"A high quality photo of a dog"`	66.55	0.371	26.48	55.91	0.249	29.65	76.45	0.394	23.58	67.75	0.354	26.10
`"A high quality photo of a face"`	60.41	0.325	26.74	52.33	0.239	29.69	77.32	0.403	23.60	68.83	0.352	26.20
Proposed	**58.73**	**0.317**	26.68	**51.40**	**0.233**	29.69	66.96	**0.386**	23.57	66.82	**0.314**	**26.29**
PALI prompts from $\boldsymbol{y}$	61.33	0.329	**26.81**	54.34	0.249	29.76	68.28	0.388	23.57	69.55	0.355	26.26
PALI prompts from $\boldsymbol{x}$	60.73	0.322	26.76	52.06	0.238	29.75	**66.55**	0.387	23.57	**64.00**	0.348	26.17

Table 6.1: Difference in restoration performance using LDPS on SR×8 task with varying text prompts. Proposed: text embedding optimized without access to ground truth. PALI prompts from $\boldsymbol{x}$/$\boldsymbol{y}$: captions are generated with PALI [22] from $\boldsymbol{x}$: ground truth clean images / $\boldsymbol{y}$: degraded images. The former can be considered an empirical upper bound.

6.1.1 Solving inverse problems with LDMs

Given access to some measurement

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n},\ \boldsymbol{A} \in \mathbb{R}^{m \times n},\ \boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \sigma_y^2 \boldsymbol{I}_m) \tag{6.1}$$

where $\boldsymbol{A}$ is the forward operator and $\boldsymbol{n}$ is additive white Gaussian noise, the task is retrieving $\boldsymbol{x} \in \mathbb{R}^n$ from $\boldsymbol{y} \in \mathbb{R}^m$. As the problem is ill-posed, a natural way to solve it is to perform posterior sampling $\boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{y})$ by defining a suitable prior $p(\boldsymbol{x})$. In DIS, DMs (i.e. denoisers) act as the implicit prior with the use of the score function. The objective of solving inverse problems is to provide a restoration that is as close as possible to the ground truth given the measurement, whether we are targeting to minimize the distortion or to maximize the perceptual quality [9, 36].

Earlier methods utilized an alternating projection approach, where hard measurement constraints are applied in-between the denoising steps whether in pixel space [77, 156] or measurement space [155, 31]. Distinctively, projection in the spectral space via singular value decomposition (SVD) to incorporate measurement noise has been developed [83, 82]. Subsequently, methods that aim to approximate the gradient of the log posterior in the diffusion model context have been proposed [26, 152], expanding the applicability to nonlinear problems. Broadening the range even further, methods that aim to solve blind [25, 117], 3D [28, 100], and unlimited resolution problems [170] were introduced. More recently, methods leveraging diffusion score functions within variational inference to solve inverse imaging has been proposed [113, 47]. Notably, all the aforementioned methods utilize *pixel-domain* DMs. Orthogonal to this direction, some of the recent works have shifted their attention to using *latent* diffusion models [139, 149, 58], a direction that we follow in this work.

In fact, inverse solvers can be directly linked to posterior sampling from $p(\boldsymbol{x}_0|\boldsymbol{y})$, which can be achieved by modifying Eq. (2.16) with

$$d\boldsymbol{x}_t = -t\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{y})\, dt = \frac{\boldsymbol{x}_t - \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}]}{t}\, dt. \tag{6.2}$$

Here, $\log p(\boldsymbol{x}_t|\boldsymbol{y}) = \log p(\boldsymbol{x}_t) + \log p(\boldsymbol{y}|\boldsymbol{x}_t)$. However, as $\log p(\boldsymbol{y}|\boldsymbol{x}_t)$ is intractable, DPS [26] proposes to approximate it with $\log p(\boldsymbol{y}|\boldsymbol{x}_t) \simeq \log p(\boldsymbol{y}|\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t])$, whose approximation error can be quantified and bounded by the Jensen gap.

This idea was recently extended to LDMs in a few recent works [139, 58], which consider the following

straightforward extension image domain DPS [26] as the baselne.

$$\begin{aligned}\nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{y}|\boldsymbol{z}_t) &\simeq \nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{y}|\mathcal{D}_{\boldsymbol{\varphi}}(\mathbb{E}[\boldsymbol{z}_0|\boldsymbol{z}_t])) \\ &= \nabla_{\boldsymbol{z}_t} \|\boldsymbol{y} - \mathcal{D}_{\boldsymbol{\varphi}}(\hat{\boldsymbol{z}}_0)\|_2^2/\sigma_y^2, \end{aligned} \tag{6.3}$$

with $\hat{\boldsymbol{z}}_0 := \mathbb{E}[\boldsymbol{z}_0|\boldsymbol{z}_t]$, leading the following latent update:

$$\boldsymbol{z}_{t-1} = \mathrm{DDIM}(\boldsymbol{z}_t) - \rho \nabla_{\boldsymbol{z}_t} \|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_{\varphi}(\hat{\boldsymbol{z}}_0)\|_2, \tag{6.4}$$

where ρ is the step size, and $\mathrm{DDIM}(\cdot)$ denotes a single step of DDIM (or DDPM in general) sampling. We refer to the sampler that uses the approximation in Eq. (6.3) as Latent DPS (LDPS) henceforth.

However, the crucial component that delineates LDM is the existence of VAE. When naively using the LDPS in Eq. (6.4), the decoder introduces a significant amount of error especially when the estimated clean latent $\hat{\boldsymbol{z}}_0^{(\mathcal{C})}$ falls off the manifold of the clean latents. To address this, [139] proposed Posterior Sampling using Latent Diffusion (PSLD) to regularize the update steps on the latent so that the clean latents are led to the fixed point of the successive application of decoding-encoding. Formally, omitting the dependence on $\mathcal{C}$, they use the following gradient step

$$\begin{aligned}\nabla_{\boldsymbol{z}_t} \log(\boldsymbol{y}|\boldsymbol{z}_t) \simeq \nabla_{\boldsymbol{z}_t} (&\|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_{\boldsymbol{\varphi}}(\hat{\boldsymbol{z}}_0)\|_2^2 + \\ &\lambda \|\hat{\boldsymbol{z}}_0 - \mathcal{E}_{\boldsymbol{\phi}}(\mathcal{D}_{\boldsymbol{\varphi}}(\hat{\boldsymbol{z}}_0))\|_2^2), \end{aligned} \tag{6.5}$$

where the additional regularization term weighted by λ leads $\hat{\boldsymbol{z}}_0$ towards the fixed point. On the other hand, [58] extends LDPS by using history updates as in Adam [88]. Concurrent work by [136] proposes to match higher-order moments of Tweedie.

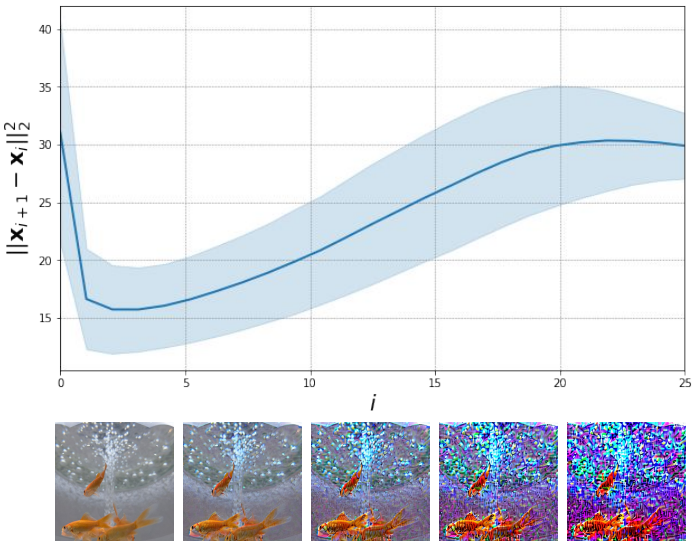


Figure 6.1: Fixed point analysis: $\mu \pm \sigma$ plotted by successive application of encoding-decoding.

The adoption of a regularized version, as indicated in Eq. (6.5), over the baseline formulation Eq. (6.3) presents a compromise between maintaining data fidelity and ensuring the stability of the VAE. This often leads to a decline in performance, particularly in scenarios with low SNR, as will be demonstrated in subsequent experimental results[2]. Furthermore, most existing works in the literature that aim for LDIS, to the best of our knowledge, neglect the

[2]Also see Fig. 6.1, where it can be seen that repeatedly applying encoding-decoding steps yields diverging results, regardless of using "glue" steps introduced in PSLD [139].

use of text embedding by resorting to the use of null text embedding $\mathcal{C}_\varnothing$. There exists one concurrent work [85] which uses a *fixed* target text while adapting the null text in CFG to emphasize the target text conditioning when solving inverse problems. Our method is orthogonal to [85] as our aim is to automatically find the ambiguous text embedding that best describes the image, rather than guide the result towards a specific mode described by the target text.

6.1.2 Prompt-tuning inverse problem solver

In modern language models and vision-language models, *prompting* is a standard technique [130, 14] to guide the large pre-trained models to solve downstream tasks. As it has been found that even slight variations in the prompting technique can lead to vastly different outcomes [93], prompt tuning (learning) has been introduced [145, 182], which defines a *learnable* context vector to optimize over. It was shown that by only optimizing over the continuous embedding vector while maintaining the model parameters fixed, one can achieve a significant performance gain. In the context of DMs, prompt tuning has been adopted for personalization [52, 116], where one defines a special token to embed a specific concept with only a few images.

Inspired by this, we are interested in the prompt optimization in LDIS. In the context of LDIS,

$$\arg\min_{\boldsymbol{x},\boldsymbol{c}} \mathcal{L}(\boldsymbol{x},\boldsymbol{c}) \equiv \arg\min_{\boldsymbol{z},\boldsymbol{c}} \mathcal{L}(\mathcal{D}_{\boldsymbol{\varphi}}(\boldsymbol{z}),\boldsymbol{c}) \tag{6.6}$$

where the first equation follows from $\boldsymbol{x} = \mathcal{D}_{\boldsymbol{\varphi}}(\boldsymbol{z})$ in the deterministic decoder mapping of VAE, where $\boldsymbol{c}$ is the text embedding and the loss $\mathcal{L}$ will be explained in more detail in subsequent session. It is easy to see that

$$\arg\min_{\boldsymbol{z},\boldsymbol{c}} \mathcal{L}(\mathcal{D}_{\varphi}(\boldsymbol{z}),\boldsymbol{c}) \leq \arg\min_{\boldsymbol{z}} \mathcal{L}(\mathcal{D}_{\varphi}(\boldsymbol{z}),\boldsymbol{c}=\mathcal{C}_\varnothing), \tag{6.7}$$

where $\mathcal{C}_\varnothing$ is the text embedding from the null text prompt. Notably, by keeping one of the variables fixed, we are optimizing for the *upper bound* of the objective that we truly wish to optimize over. It would be naturally beneficial to optimize the LHS of Eq. (6.7), rather than the RHS used in the previous methods.

To see Eq. (6.7) in effect, we conduct two canonical experiments with 256 test images of FFHQ [80] and ImageNet [37]: super-resolution (SR) of scale $\times 8$ and inpainting with 80% of the pixels randomly dropped, using the LDPS algorithm. Keeping all the other hyper-parameters fixed, we only vary the text condition for the diffusion model. In addition to using a general text prompt, we use PALI [22] to provide captions from the ground truth images ($\boldsymbol{x}$) and from the measurements ($\boldsymbol{y}$) and use them when running LDPS. In Table 6.1, we first see that simply varying the text prompts can lead to dramatic difference in the performance. For instance, we see an increase of over **10 FID** when we use the text prompts from PALI for the task of $\times 8$ SR on ImageNet. In contrast, using the prompts generated from $\boldsymbol{y}$ often degrades the performance (e.g. inpainting) as the correct captions cannot be generated. Indeed, from the table, we see that by applying our prompt tuning approach, we achieve a large performance gain, sometimes even outperforming the PALI captions which has full access to the ground truth when attaining the text embeddings. From this motivating example, it is evident that additionally optimizing for $\boldsymbol{c}$ would bring us gains that are orthogonal to the development of the solvers [139, 58, 149], a direction which will be explored in this paper.

To effectively utilize the Latent Diffusion Inverse Solver (LDIS) with prompt optimization, it is crucial to ensure two key criteria: 1) consistency with respect to the measurements, and 2) the feasibility of the latent as per the LDM. Our approach diverges from conventional regularization strategies, such as PSLD [139]. Instead, we

Algorithm 10 P2L

Require: $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}, \boldsymbol{z}_T, \boldsymbol{y}, \mathcal{C}, T, K, \gamma, \lambda_{\mathcal{D}}$

for $t = T$ **to** 1 **do**
  $\mathcal{C}_t^* \leftarrow \textsc{OptimizeEmb}(\boldsymbol{z}_t, \boldsymbol{y}, \mathcal{C}_t^0, K)$ ▷ Sec. 6.1.3
  $\hat{\boldsymbol{\epsilon}}_t \leftarrow \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, \mathcal{C}_t^*)$
  $\hat{\boldsymbol{z}}_{0|t} \leftarrow (\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}_t)/\sqrt{\bar{\alpha}_t}$
  if $(t \mod \gamma) = 0$ **then**
    $\hat{\boldsymbol{x}}_0 \leftarrow \arg\min_{\boldsymbol{x}_0} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_0\|_2^2 + \lambda\|\boldsymbol{x}_0 - \mathcal{D}_{\varphi}(\hat{\boldsymbol{z}}_{0|t})\|_2^2$
    $\tilde{\boldsymbol{z}}_{0|t} \leftarrow \mathcal{E}_{\boldsymbol{\phi}}(\hat{\boldsymbol{x}}_0)$ ▷ Sec. 6.1.4
  else
    $\tilde{\boldsymbol{z}}_{0|t} \leftarrow \hat{\boldsymbol{z}}_{0|t}$
  end if
  $\boldsymbol{z}'_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}}\tilde{\boldsymbol{z}}_{0|t} + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\boldsymbol{\epsilon}}_t$
  $\boldsymbol{z}_{t-1} \leftarrow \boldsymbol{z}'_{t-1} - \rho_t \nabla_{\boldsymbol{z}_t}\|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_{\boldsymbol{\varphi}}(\hat{\boldsymbol{z}}_{0|t})\|$ ▷ Sec. 6.1.5
  $\mathcal{C}_{t-1}^{(0)} \leftarrow \mathcal{C}_t^*$
end for
return $\boldsymbol{x}_0 \leftarrow \mathcal{D}_{\boldsymbol{\varphi}}(\boldsymbol{z}_0)$

Algorithm 11 Prompt tuning

function $\textsc{OptimizeEmb}(\boldsymbol{z}_t, \boldsymbol{y}, \mathcal{C}_t^{(0)}, K)$
  for $k = 1$ **to** K **do**
    $\hat{\boldsymbol{\epsilon}}_t \leftarrow \boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\boldsymbol{z}_t, \mathcal{C}_t^{(k-1)})$
    $\hat{\boldsymbol{z}}_{0|t} \leftarrow (\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}_t)\sqrt{\bar{\alpha}_t}$
    $\hat{\boldsymbol{z}}'_{0|t} \leftarrow \hat{\boldsymbol{z}}_{0|t} - \rho\nabla_{\hat{\boldsymbol{z}}_{0|t}}\|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_{\boldsymbol{\varphi}}(\hat{\boldsymbol{z}}_{0|t})\|$
    $\mathcal{L}_t \leftarrow \|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_{\boldsymbol{\varphi}}(\hat{\boldsymbol{z}}'_{0|t}, \mathcal{C}_t^{(k-1)})\|_2^2$
    $\mathcal{C}_t^{(k)} \leftarrow \mathcal{C}_t^{(k-1)} - \texttt{AdamGrad}(\mathcal{L}_t)$
  end for
  return $\mathcal{C}_t^* \leftarrow \mathcal{C}_t^{(K)}$
end function

base our formulation on Eq. (6.7), which offers a more direct route to achieving these objectives:

$$\min_{\boldsymbol{z}\in P(\boldsymbol{z}|\boldsymbol{y})} \min_{\mathcal{C}} \quad \|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_{\varphi}(\boldsymbol{z}^{(\mathcal{C})})\|^2 \tag{6.8}$$

$$\text{subject to} \quad \boldsymbol{z} \in F_X \tag{6.9}$$

where $P(\boldsymbol{z}|\boldsymbol{y})$ denotes the posterior distribution of $\boldsymbol{z}$ given the measurement condition $\boldsymbol{y}$ and F_X denotes the set of latent that can be represented by some image x:

$$F_X = \{\boldsymbol{z} | \boldsymbol{z} = \mathcal{E}_{\phi}^{\mu}(\boldsymbol{x}) \text{ for some } \boldsymbol{x}\}$$

A key contribution of our study is the demonstration that the optimization problem involving prompt, latent, and pixel values can be effectively addressed through alternating minimization, as explained in the following sections. We summarize our alternating sampling method in Algorithm 10 and Algorithm 11, based on DDIM sampling, with standard noise schedule notations adopted from [62].

The intuition of the overall algorithm is that by incorporating the text conditioning automatically, ambiguities arising from the natural ill-posedness of the inverse problems can be mitigated. Further, artifacts that often arise from naive latent space optimization can be corrected by leveraging the VAE during inverse problem-solving. Brief overview of the method structure:

1. (Sec. 6.1.3) Prompt embedding optimization through fidelity loss minimization

2. (Sec. 6.1.4) Latent update via LDPS step with the optimized prompt

3. (Sec. 6.1.5) Latent correction. This involves decoding, enforcing data consistency in the pixel space, and re-encoding

6.1.3 Prompt tuning

To update the prompt, we address the inner optimization challenge presented in Eq. (6.8), which involves identifying a suitable latent that aligns with $z \in P(\boldsymbol{z}|\boldsymbol{y})$. This process is akin to the approach used in decomposed diffusion sampling (DDS) as described in [27]. It entails minimizing the loss detailed in Eq. (6.8), starting from the denoised latent $\hat{\boldsymbol{z}}_{0|t}$. As a result, we obtain $\hat{\boldsymbol{z}}'_{0|t}$ as a first order approximation of $\mathbb{E}[\boldsymbol{z}_0|\boldsymbol{z}_t, \boldsymbol{y}, \mathcal{C}]$, by adjusting $\hat{\boldsymbol{z}}_{0|t}$ and incorporating data consistency, as outlined in Line 5 of Algorithm 11. Now for the updated latent $\hat{\boldsymbol{z}}'_{0|t}$,we should solve the inner optimization problem of Eq. (6.8), leading to the following optimization:

$$\mathcal{C}_t^* = \arg\min_{\mathcal{C}} \|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_{\boldsymbol{\varphi}}(\hat{\boldsymbol{z}}_{0|t}^{(\mathcal{C})})\|_2^2 \tag{6.10}$$

We found that this alternating minimization should be solved multiple times to have meaningful update of the problem. This corresponds to the OPTIMIZEEMB in Algorithm 10, with details of the optimization function in Algorithm 11.

6.1.4 Enforcing data fidelity

For a given optimized prompt $\mathcal{C}_t^*$, a straightforward extension of the LDPS for latent update $\boldsymbol{z}_{t-1}$ in Eq. (6.4) is

$$\boldsymbol{z}_{t-1} = \mathrm{DDIM}(\boldsymbol{z}_t) - \rho\nabla_{\boldsymbol{z}_t}\|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_{\varphi}(\hat{\boldsymbol{z}}_0^{(\mathcal{C}_t^*)})\|_2^2, \tag{6.11}$$

where $\hat{\boldsymbol{z}}_0^{(\mathcal{C}_t)} := \mathbb{E}[\boldsymbol{z}_0|\boldsymbol{z}_t, \mathcal{C}_t^*]$ is the prompt conditioned posterior mean. Here, $\mathrm{DDIM}(\boldsymbol{z}_t)$ can be equivalently represented by the denosing step through Tweedie's formula:

$$\hat{\boldsymbol{z}}_{0|t} := (\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}_t)/\sqrt{\bar{\alpha}_t} \tag{6.12}$$

followed by the noising step:

$$\mathrm{DDIM}(\boldsymbol{z}_t) = \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{z}}_{0|t} + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\boldsymbol{\epsilon}}_t \tag{6.13}$$

where $\hat{\boldsymbol{\epsilon}}_t := \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, \mathcal{C}_t^*)$, as shown in [27]. These data fidelity enforcing steps are presented in line 3-4,9-12 of Algorithm 10.

6.1.5 Enforcing latent feasibility

However, the aforementioned data fidelity enforcing steps do not consider the latent constraint Eq. (6.9). Specifically, without considering our constraint, we see in Fig. 6.2 that artifacts arise, and this cannot be fully mitigated by leveraging the regularizations proposed in PSLD [139]. In this section, we show that this constraint can be easily enforced by incorporating the VAE prior.

Specifically, inspired by the regularization term in PSLD in Eq. (6.5), we consider the following loss, which is

the maximum a posteriori (MAP) objective under the VAE prior in Eq. (2.33) with isotropic covariance [55].

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{z}) = \|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_{\varphi}(\boldsymbol{z})\|_2^2 + \zeta\|\boldsymbol{z} - \mathcal{E}_{\boldsymbol{\phi}}^{\boldsymbol{\mu}}(\boldsymbol{x})\|_2^2, \tag{6.14}$$

where ζ absorbs the weighting caused by the variance of respective terms. Instead of enforcing the *hard* constraint between the $\boldsymbol{x}$ and $\boldsymbol{z}$ in the form of $\boldsymbol{x} = \mathcal{D}_{\boldsymbol{\varphi}}(\boldsymbol{z})$ as in [139] that can introduce the trade-off between data consistency and the stability of latent, our main goal is to enforce a *soft* constraint by splitting the variables similar in spirit to the alternating direction method of multipliers (ADMM) [11].

Namely, using the variable splitting $\boldsymbol{x} = \mathcal{D}_{\varphi}(\boldsymbol{z})$, the optimization problem with respect to $\boldsymbol{x}$ becomes

$$\begin{aligned}\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \zeta\|\boldsymbol{z} - \mathcal{E}_{\boldsymbol{\phi}}^{\boldsymbol{\mu}}(\mathcal{D}_{\varphi}(\boldsymbol{z}))\|_2^2 \\ + \lambda\|\boldsymbol{x} - \mathcal{D}_{\varphi}(\boldsymbol{z}) + \boldsymbol{\eta}\|_2^2.\end{aligned} \tag{6.15}$$

where $\boldsymbol{\eta}$ denotes the dual variable in ADMM. Since we consider using only a single step ADMM update for each diffusion sampling step, we set dual variable $\boldsymbol{\eta}$ as a zero vector and do not consider its update. This leads to

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x} - \mathcal{D}_{\varphi}(\boldsymbol{z})\|_2^2. \tag{6.16}$$

Solving for Eq. (6.16) is performed using conjugate gradient (CG) with the *clean* latent $\hat{\boldsymbol{z}}_{0|t}$ obtained through the Tweedie's formula Eq. (6.12), leading to the following update:

$$\hat{\boldsymbol{x}}_0 = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x} - \mathcal{D}_{\varphi}(\hat{\boldsymbol{z}}_{0|t})\|_2^2 \tag{6.17}$$

as presented in line 3-6 of Algorithm 10. The resulting optimization problem in Eq. (6.17) is indeed a pixel-domain proximal update from $\mathcal{D}_{\varphi}(\hat{\boldsymbol{z}}_{0|t})$, which can be interpreted as enforcing the data fidelity in the pixel domain under the regularization from latent feasiblity.

Subsequently, using the encoder approximation and setting $\boldsymbol{z} = \mathcal{E}_{\boldsymbol{\phi}}(\boldsymbol{x})$ with $\boldsymbol{\eta} = \boldsymbol{0}$, the optimization problem with respect to $\boldsymbol{z}$ reads

$$\min_{\boldsymbol{z}} \|\boldsymbol{y} - \boldsymbol{A}\mathcal{D}_{\boldsymbol{\varphi}}\mathcal{E}_{\boldsymbol{\phi}}^{\boldsymbol{\mu}}(\boldsymbol{x})\|_2^2 + \zeta\|\boldsymbol{z} - \mathcal{E}_{\boldsymbol{\phi}}^{\boldsymbol{\mu}}(\boldsymbol{x})\|_2^2.$$

For a given pixel-domain update $\hat{\boldsymbol{x}}_0$ from Eq. (6.17), the corresponding latent update then has the closed-form solution

$$\hat{\boldsymbol{z}}_{0|t} = \mathcal{E}_{\boldsymbol{\phi}}^{\boldsymbol{\mu}}(\hat{\boldsymbol{x}}_0) \tag{6.18}$$

Note that by Eq. (6.18), we guarantee that the clean latents stay on the *range space* of the encoder, automatically satisfying the constraint in Eq. (6.9). For this reason, we often denote the method proposed in this section simply as "projection" to the constraint set.

In practice, we choose to apply Eq. (6.17) and Eq. (6.18) every few iteration to control dramatic changes in the sampling, and to save computation. Nevertheless, solving Eq. (6.17) requires access to $\boldsymbol{A}^\top$, which is often non-trivial to define. Contrarily, our `jax` implementation enables defining $\boldsymbol{A}^\top$ through `jax.vjp`. Upon implementing Eq. (6.17) and Eq. (6.18), we reintroduce a data consistency step, as demonstrated in lines 11-12 of Algorithm 10. This step is to ensure that the process does not deviate from data consistency.

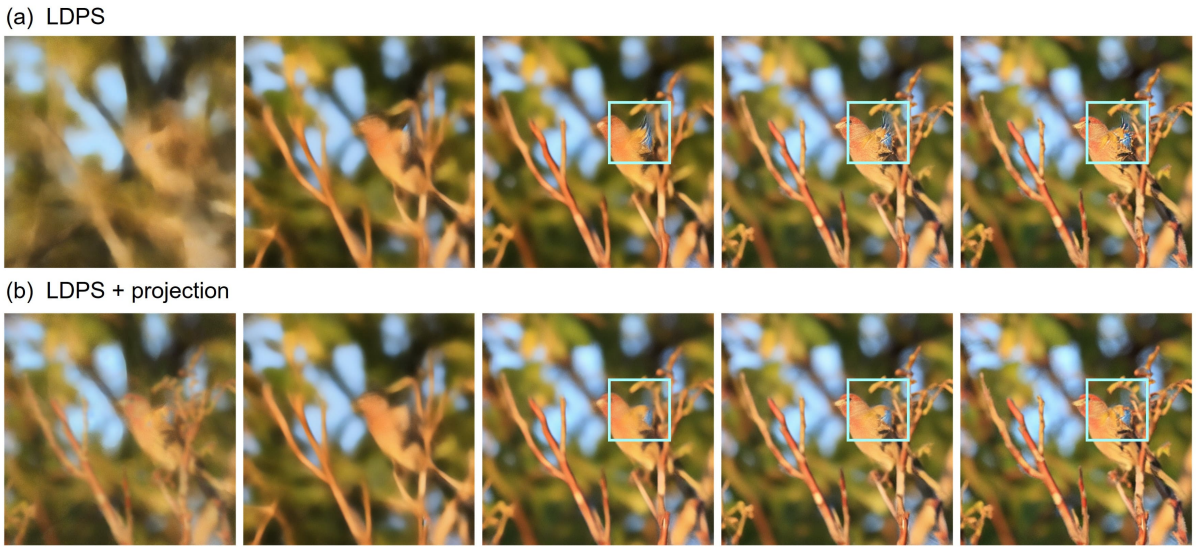


Figure 6.2: Evolution of DIS while solving SR×8 with (a) LDPS, (b) LDPS + projection. Using projection steps help mitigate the artifacts.

6.1.6 Targetting arbitrary resolution

Another important contribution of this work is its scalability to arbitrary resolution with large image size. Despite its fully convolutional nature, as SD was trained with 64×64 latents ($\leftrightarrow 512\times512$ images), the performance degrades when we aim to deal with larger dimensions, again due to train-test time discrepancy. Several works aimed to mitigate this issue by processing the latents with strided patches [4, 72, 167] that increases the computational burden by roughly $\mathcal{O}(n^2)$. In contrast, we show that our approach using the projection step by simply running Alg. 10, used *without* any patch processing, can outperform previous methods that rely on patches, resulting in significantly improved image quality and faster inference speed. This is because when given a latent that stays within the range space of the encoder thanks to Eq. (6.18), the decoder is able to produce a high-quality image directly even when the input size is larger than 64× 64.

Guidance on hyperparameter selection P2L, with prompt embedding as an additional variable to optimize over, has more hyperparameters than standard DIS. Here, we provide a solid choice that works well across most experiments. 1) For optimizing prompt embedding, 1 5 iterations (K) with a learning rate of $1e-4$ yields stable performance. We observe setting too high values of K or learning rate leads to overfitting, while setting them too small yields marginal improvements. 2) One can reliably choose GD with static step size of 1.0 for LDPS update, as advised in many previous works [26, 139]. 3) projection works when applied every 3-5 steps (γ), while the value of λ matters less and can be freely chosen between 0.1 1.0 with negligible difference in the performance. When applying projection too often, artifacts arise.

6.1.7 Experiments

Datasets, Models We consider two different well-established datasets: 1) FFHQ 512×512 [80], and 2) ImageNet 512×512 [37]. For the former, we use the first 1000 images for testing, similar to [26]. For the latter, we choose 1k images out of 10k test images provided in [140] by interleaved sampling, i.e. using images of index 0, 10, 20, etc. after ordering by name. For the latent diffusion model, we choose SD v1.4 pre-trained on the LAION dataset for all the experiments, including the baseline comparison methods based on LDM. As there is no publicly available image diffusion model that is trained on an identical dataset, we choose ADM [39] trained on ImageNet 512×512

Figure 6.3: Inverse problem solving results on ImageNet 512×512 test set. Row 1: SR$\times$8, Row 2: gaussian deblurring, Row 3: motion deblurring, row 4: inpainting.

data as the universal prior when implementing baseline pixel-domain DIS. Note that this discrepancy may lead to an unfair advantage in the performance for evaulation on ImageNet, and an unfair disadvantage in the performance when evaluating on FFHQ. All experiments were done on NVIDIA A100 40GB GPUs.

Inverse Problems We test our method on the following degradations: 1) Super-resolution from $\times 8$ average-pooling, 2) Inpainting from 10-20% free-form masking as used in [140], 3) Gaussian deblurring from an image convolved with a 61×61 size Gaussian kernel with $\sigma = 3.0$, 4) Motion deblurring from an image convolved with a 61×61 motion kernel that is randomly sampled with intensity 0.5[3], following [26]. For all degradations, we include mild additive white Gaussian noise with $\sigma_y = 0.01$.

Evaluation As the main objective of this study is to improve the performance of LDIS, we mainly focus our evaluation on the comparison against the current SOTA LDIS: we compare against LDPS, GML-DPS [139], PSLD [139], and LDIR [58]. We additionally compare against TRreg [85] to emphasize that the aim of the works are different. All LDIS including the proposed P2L use 1000 NFE DDIM sampling with $\eta = 0.0$[4], with the exception of TReg, which uses 200 NFE DDIM sampling. Using higher NFE did not help in improving sample quality. We additionally compare against SOTA pixel-domain DIS: DPS [26], Diff-PIR [183], DDS [27], and ΠGDM [152]. For DPS, we use 1000 NFE DDIM sampling. For Diff-PIR, DDS, and ΠGDM, we use 100 NFE DDIM sampling. We choose the optimal η values for these algorithms through grid-search. We perform a quantitative evaluation with standard metrics: PSNR, FID, and LPIPS.

Comparison against baseline In all of the inverse problems that we consider in the paper, our method outperforms all the baselines by quite a large margin in terms of perceptual quality, measured by FID and LPIPS, while keeping

[3] https://github.com/LeviBorodenko/motionblur

[4] The parameter η indicates the stochasticity of the sampler. $\eta = 0.0$ leads to deterministic PF-ODE.

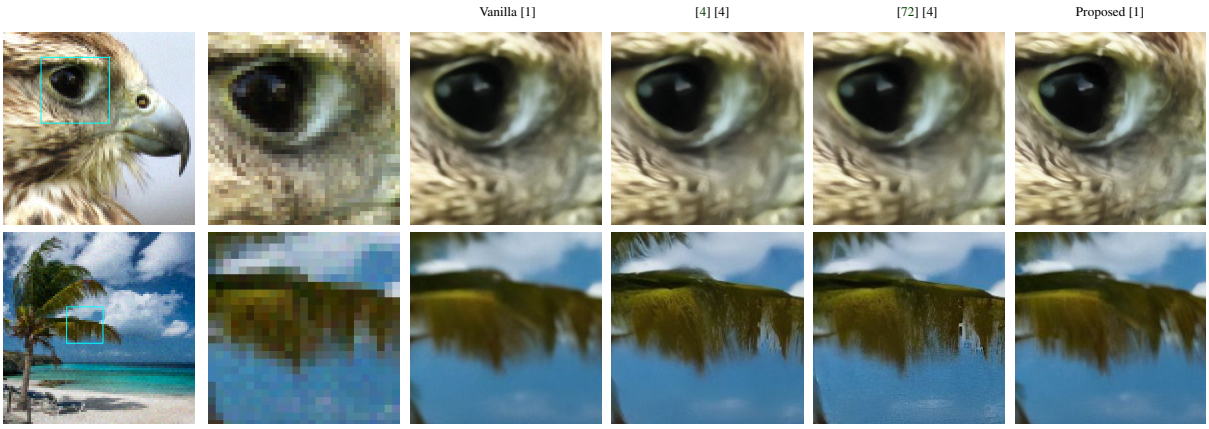


Figure 6.4: Results on $\times 8$ SR on DIV2K validation set of 768×768 resolution. [Diffusion NFE per denoising step]. Vanilla and proposed process the latent as a whole.

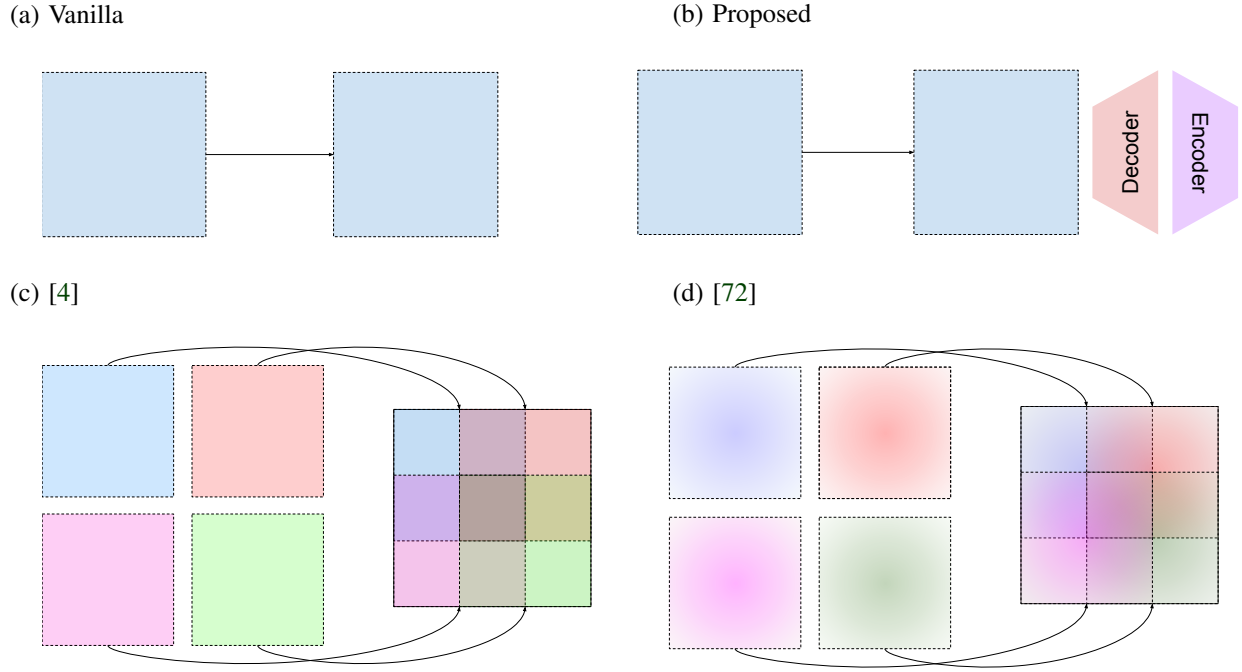


Figure 6.5: Method comparison for processing higher resolution images in the latent space.

	SR (×8)			Deblur (motion)			Deblur (gauss)			Inpaint		
Method	FID↓	LPIPS↓	PSNR↑	FID↓	LPIPS↓	PSNR↑	FID↓	LPIPS↓	PSNR↑	FID↓	LPIPS↓	PSNR↑
P2L (ours)	**51.81**	**0.386**	**23.38**	**54.11**	**0.360**	**24.79**	**39.10**	**0.325**	25.11	**32.82**	**0.229**	<u>21.99</u>
LDPS	61.09	0.475	<u>23.21</u>	71.12	0.441	23.32	48.17	0.392	24.91	46.72	0.332	21.54
GML-DPS [139]	60.36	<u>0.456</u>	<u>23.21</u>	59.08	0.403	24.35	<u>45.33</u>	0.377	**25.44**	47.30	0.294	21.12
PSLD [139]	60.81	0.471	23.17	59.63	0.398	24.21	45.44	<u>0.376</u>	<u>25.42</u>	<u>40.57</u>	<u>0.251</u>	20.92
LDIR [58]	63.46	0.480	22.23	88.51	0.475	21.37	72.10	0.506	22.45	50.65	0.313	**23.28**
TReg [85]	104.3	0.520	18.97	102.97	0.501	19.06	117.3	0.455	16.84	77.76	0.349	14.98
DDS [27]	203.2	1.213	12.72	84.67	0.925	14.52	70.51	0.835	16.58	60.18	0.354	17.03
DPS [26]	54.61	0.544	20.70	71.99	0.599	19.62	98.33	0.910	15.05	71.70	0.360	15.15
DiffPIR [183]	488.3	1.182	13.44	87.04	0.622	19.32	79.31	0.755	20.55	45.97	0.300	20.11
ΠGDM [152]	<u>53.00</u>	0.490	21.08	75.35	0.682	18.66	70.26	0.797	21.96	65.75	0.322	16.84

Table 6.2: Quantitative evaluation (PSNR, LPIPS, FID) of inverse problem solving on ImageNet 512×512-1k validation dataset. **Bold**: best, <u>underline</u>: second best. Methods that are not LDM-based are shaded in gray.

Method	SR (×8)			Deblur (motion)			Deblur (gauss)			Inpaint		
	FID↓	LPIPS↓	PSNR↑	FID↓	LPIPS↓	PSNR↑	FID↓	LPIPS↓	PSNR↑	FID↓	LPIPS↓	PSNR↑
P2L (ours)	**31.23**	**0.290**	28.55	28.34	**0.302**	**27.23**	**30.62**	**0.299**	26.97	**26.27**	**0.168**	25.29
LDPS	36.81	0.292	**28.78**	58.66	0.382	26.19	45.89	0.334	27.82	46.10	0.311	23.07
GML-DPS [139]	41.65	0.318	28.50	47.96	0.352	27.16	42.60	0.320	**28.49**	36.31	0.208	23.10
PSLD [139]	36.93	0.335	26.62	47.71	0.348	27.05	41.04	0.320	28.47	35.01	0.207	23.10
LDIR [58]	36.04	0.345	25.79	**24.40**	0.376	24.40	35.61	0.341	25.75	37.23	0.250	**25.47**
DDS [27]	262.0	1.278	13.01	88.70	1.014	14.68	74.02	0.932	17.03	113.6	0.421	17.92
DPS [26]	47.65	0.340	21.81	65.91	0.601	21.11	100.2	0.983	15.71	137.7	0.692	15.35
DiffPIR [183]	141.1	1.266	13.80	72.02	0.664	21.03	69.15	0.751	22.27	33.92	0.238	24.91
ΠGDM [152]	42.07	0.311	22.05	60.08	0.531	21.08	70.32	0.788	21.99	140.6	0.738	16.83

Table 6.3: Quantitative evaluation (PSNR, LPIPS, FID) of inverse problem solving on FFHQ 512×512-1k validation dataset. **Bold**: best, underline: second best. Methods that are not LDM-based are shaded in gray.

Design components			FFHQ				ImageNet			
			SR×8		Inpaint ($p = 0.8$)		SR×8		Inpaint ($p = 0.8$)	
Projection	Γ	Prompt tuning	FID↓	PSNR↑	FID↓	PSNR↑	FID↓	PSNR↑	FID↓	PSNR↑
✗	✗	✗	61.16	26.49	52.34	**29.78**	78.68	23.49	70.87	26.20
✗	✗	✓	58.73	**26.68**	51.40	29.69	76.40	**23.52**	67.06	26.32
✓	✗	✗	55.91	26.37	48.71	29.68	74.22	23.16	66.92	26.08
✓	✓	✗	55.68	26.43	47.76	29.70	74.01	23.32	65.45	26.29
✓	✓	✓	**52.96**	26.64	**46.92**	29.63	**70.08**	23.48	**59.26**	26.12

Table 6.4: Ablation studies on the design components

σ_y	Γ	PSNR	FID
0.0	glue	26.51	54.69
	Ours	**26.80**	**54.58**
0.01	glue	26.39	56.47
	Ours	**26.43**	**55.68**
0.05	glue	23.86	68.99
	Ours	**24.92**	**65.90**

Table 6.5: Choice of Γ

the distortion at a comparable level against the current state-of-the-art methods. Especially, we see about 10 FID decrease in deblurring and inpainting tasks compared to the runner up in both FFHQ and ImageNet dataset (See Tables 6.3,6.2). The superiority can also be clearly seen in Fig. 6.3, where P2L achieves stable, high-quality reconstruction throughout all tasks. Results from both LDPS and PSLD often contain local grid-like artifacts (Red boxes in Figures) and are blurry. With P2L, the restored images are sharpened while the artifacts are effectively removed. LDIR are less prone to artifacts owing to the smoothed history gradient updates, but often results in unrealistic textures and deviations from the measurement, which is also reflected in having the lowest PSNR among the LDIS-class methods. In contrast, P2L is free from such drawbacks even when leveraging Adam-like gradient update steps. It should be noted that the compute time for P2L linearly increases as we increase the number of training iterations for the text embedding. The compute time for $K = 0$ is similar to other LDIS baselines, but it becomes slower if K becomes larger. Devising a more time-efficient way to perform text embedding optimization is thus a promising future research direction.

One rather surprising finding is the heavy downgrade in the performance for DIS methods. Even on in-distribution ImageNet test data, methods such as DPS and DiffPIR become very unstable. This can be attributed to the generative prior being poor: directly training DMs on high-resolution images often result in poor performance[5]. This observation again points to the importance of developing methods that can leverage foundation models when aiming for general domain higher-resolution data. As a final note, we believe that the compromise in PSNR is related to the imperfectness of the VAE used in SD v1.4[6], and we expect such degradation to be mitigated when switching to better, larger autoencoders such as SDXL [127].

Design components In Table 6.4, we perform an ablation study on the design components of the proposed method. From the table, we confirm that prompt tuning, projection to the range space of the encoder, and performing

[5]For $\geq 512 \times 512$ resolution, either using latent diffusion or using cascaded models [141] are popular.

[6]Auto-encoding 1000 ground-truth test images result in the following metrics: FFHQ (PSNR): 29.66 ± 2.29, ImageNet (PSNR): 27.12 ± 4.38.

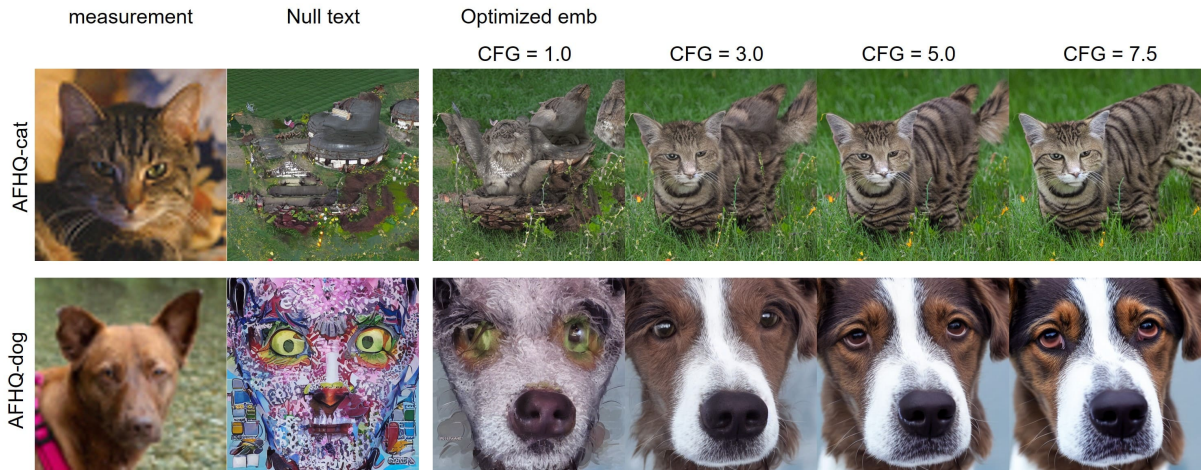


Figure 6.6: Indirect visualization of the optimized embedding through solving an inverse problem with P2L. After solving SR×8 with measurements in the first column, we perform unconditional sampling by fixing the random seed, and replacing the condition with the optimized embedding by varying the CFG.

proximal update step (denoted as $\mathbf{\Gamma}$) before the projection all contributes to the gain in the performance. It is important that these gains are synergistic, and one component does not hamper the other. Our prompt-tuning approach is robust to the variation in the hyper-parameters (learning rate, number of iterations). Specifically, among the 9 configurations that we try, only the one with 5 iterations, lr=0.001 is inferior to not using prompt tuning. In Fig. 6.2, we visualize the progress of $\mathcal{D}(\hat{\boldsymbol{z}}_0)$ through time t starting from the same random seed, comparing LDPS, PSLD, and LDPS + projection (row 4 of Tab. 6.4). Here, we see that our proposed projection approach effectively suppresses the artifacts that arise during the reconstruction process, whereas PSLD introduces additional artifacts. Furthermore, we show that our approach is also useful for targetting arbitrary resolution image restoration, as the errors accumulated by processing latents in higher dimensions can be corrected through our projection approach. Remarkably, we see that our approach often offers better results (e.g. see Fig. 6.4) than operating in strided patches [4, 72], which requires quadratic scaling of compute time.

Choice of $\mathbf{\Gamma}$ When projecting to the range space of $\mathcal{E}$, we choose to use the proximal optimization strategy in Eq. (6.17) and Eq. (6.18). Instead, one could resort to projection to the measurement subspace ("gluing" of [139]) by using $\mathbf{\Gamma}(\hat{\boldsymbol{x}}_0) = \boldsymbol{A}^\top \boldsymbol{y} + (\boldsymbol{I} - \boldsymbol{A}^\top \boldsymbol{A})\hat{\boldsymbol{x}}_0$. In Table 6.5, we compare our choice of $\mathbf{\Gamma}$ against the gluing on various noise levels on FFHQ SR×8. We see that for all noise levels, our projecton steps consistently outperform the gluing, even when $\mathbf{\Gamma}$ is applied every $\gamma = 4$ steps of reverse diffusion. Furthermore, the differences become more pronounced as we increase the noise level. The difference in the compute time between the two choices is minimal: 331.7 [s] vs 333.2 [s] measured in wall-clock time using RTX 3090 GPU per the restoration of a single image when we compare gluing vs. proximal optimization.

Visualization of the optimized prompt Although the optimized prompt during the P2L inference cannot be directly decoded as a text, we can indirectly try to visualize what the prompt has *learned* from the optimization process. If the embedding was optimized in a meaningful way, we would expect it to contain some information about the underlying image. Hence, when we use this embedding to generate samples with standard CFG, we would achieve images that are more similar to the underlying image, compared to not using this embedding. In Fig. 6.6, we verify that this is indeed the case on the SR×8 experiment on AFHQ cat and dog images.

6.2 CFG++: Manifold-constrained Classifier Free Guidance for Diffusion Models

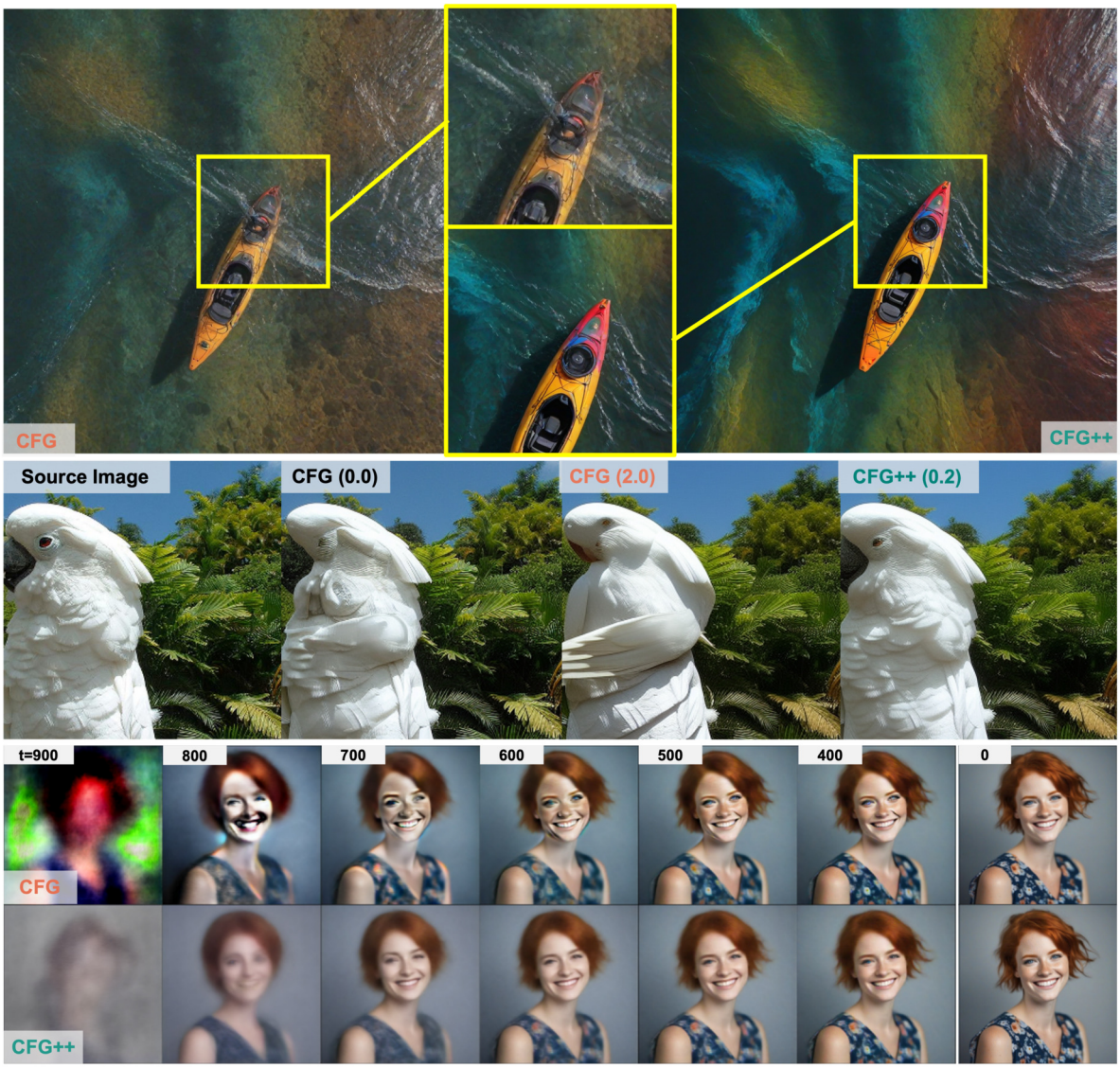


Figure 6.7: **(Top)** Comparison of T2I results by SDXL-Turbo for the prompt "kayak in the water, optical color, aerial view, rainbow". The CFG-guided image has significant artifacts, which are reduced in the CFG++ version. **(Middle)** DDIM Inversion results under CFG show noticeable artifacts at various CFG scales, which are significantly reduced by CFG++. **(Bottom)** The evolution of denoised estimates differs between CFG and CFG++. CFG exhibits sudden shifts and intense color saturation early in reverse diffusion, while CFG++ transitions smoothly from low to high-resolution.

Classifier-free guidance (CFG) [63] forms the key basis of modern text-guided generation with diffusion models [39, 135]. Nowadays, it is common practice to train a diffusion model with large-scale paired text-image data [143], so that sampling (i.e. generating) a signal (e.g. image, video) from a diffusion model can either be done unconditionally from $p_\theta(\boldsymbol{x}|\varnothing) \equiv p_\theta(\boldsymbol{x})$, or conditionally from $p_\theta(\boldsymbol{x}|\boldsymbol{c})$, where $\boldsymbol{c}$ is the text conditioning. Once trained, it seems natural that one would acquire samples from the conditional distribution by simply solving the probability-flow ODE or SDE sampling [156, 151, 79] with the conditional score function. In practice, however, it is observed that the conditioning signal is insufficient when used naively. To emphasize the guidance, one uses

Algorithm 12 Reverse Diffusion with CFG

Require: $\boldsymbol{x}_T \sim \mathcal{N}(0, \mathbf{I}_d), 0 \leq \omega \in \mathbb{R}$

for $i = T$ **to** 1 **do**
  $\hat{\boldsymbol{\epsilon}}^{\omega}_{\boldsymbol{c}}(\boldsymbol{x}_t) = \hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t) + \omega[\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_t) - \hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t)]$
  $\hat{\boldsymbol{x}}^{\omega}_{\boldsymbol{c}}(\boldsymbol{x}_t) \leftarrow (\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}^{\omega}_{\boldsymbol{c}}(\boldsymbol{x}_t))/\sqrt{\bar{\alpha}_t}$
  $\boldsymbol{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{x}}^{\omega}_{\boldsymbol{c}}(\boldsymbol{x}_t) + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\boldsymbol{\epsilon}}^{\omega}_{\boldsymbol{c}}(\boldsymbol{x}_t)$
end for
return $\boldsymbol{x}_0$

Algorithm 13 Reverse Diffusion with CFG++

Require: $\boldsymbol{x}_T \sim \mathcal{N}(0, \mathbf{I}_d), \lambda \in [0, 1]$

for $i = T$ **to** 1 **do**
  $\hat{\boldsymbol{\epsilon}}^{\lambda}_{\boldsymbol{c}}(\boldsymbol{x}_t) = \hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t) + \lambda[\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_t) - \hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t)]$
  $\hat{\boldsymbol{x}}^{\lambda}_{\boldsymbol{c}}(\boldsymbol{x}_t) \leftarrow (\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}^{\lambda}_{\boldsymbol{c}}(\boldsymbol{x}_t))/\sqrt{\bar{\alpha}_t}$
  $\boldsymbol{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{x}}^{\lambda}_{\boldsymbol{c}}(\boldsymbol{x}_t) + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t)$
end for
return $\boldsymbol{x}_0$

Algorithm 14 DDIM Inversion with CFG

Require: $\boldsymbol{x}_0, 0 \leq \omega \in \mathbb{R}$

for $i = 0$ **to** $T-1$ **do**
  $\hat{\boldsymbol{\epsilon}}^{\omega}_{\boldsymbol{c}}(\boldsymbol{x}_t) = \hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t) + \omega[\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_t) - \hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t)]$
  $\hat{\boldsymbol{x}}^{\omega}_{\boldsymbol{c}}(\boldsymbol{x}_t) \leftarrow (\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}^{\omega}_{\boldsymbol{c}}(\boldsymbol{x}_t))/\sqrt{\bar{\alpha}_t}$
  $\boldsymbol{x}_{t+1} = \sqrt{\bar{\alpha}_{t+1}}\hat{\boldsymbol{x}}^{\omega}_{\boldsymbol{c}}(\boldsymbol{x}_t) + \sqrt{1-\bar{\alpha}_{t+1}}\hat{\boldsymbol{\epsilon}}^{\omega}_{\boldsymbol{c}}(\boldsymbol{x}_t)$
end for
return $\boldsymbol{x}_T$

Algorithm 15 DDIM Inversion with CFG++

Require: $\boldsymbol{x}_0, \lambda \in [0, 1]$

for $i = 0$ **to** $T-1$ **do**
  $\hat{\boldsymbol{\epsilon}}^{\lambda}_{\boldsymbol{c}}(\boldsymbol{x}_t) = \hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t) + \lambda[\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_t) - \hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t)]$
  $\hat{\boldsymbol{x}}^{\lambda}_{\boldsymbol{c}}(\boldsymbol{x}_t) \leftarrow (\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t))/\sqrt{\bar{\alpha}_t}$
  $\boldsymbol{x}_{t+1} = \sqrt{\bar{\alpha}_{t+1}}\hat{\boldsymbol{x}}^{\lambda}_{\boldsymbol{c}}(\boldsymbol{x}_t) + \sqrt{1-\bar{\alpha}_{t+1}}\hat{\boldsymbol{\epsilon}}^{\lambda}_{\boldsymbol{c}}(\boldsymbol{x}_t)$
end for
return $\boldsymbol{x}_T$

the guidance scale $\omega > 1$, where the direction can be defined by the direction from the unconditional score to the conditional score [63].

In modern text-to-image (T2I) diffusion models, the guidance scale ω is typically set within the range of $[5.0, 30]$, referred to as the *moderately* high range of CFG guidance [20, 127]. The insufficiency in guidance also holds for classifier guidance [39, 156] so that a scale of 10 was used. While using a high guidance scale yields higher-quality images with better alignment to the condition, it is also prone to mode collapse, reduces sample diversity, and yields an inevitable accumulation of errors during the sampling process. One example is DDIM inversion [39], a pivotal technique for controllable synthesis and editing [116], where running the inversion process with $\omega > 1.0$ leads to significant compromise in the reconstruction performance [116, 166]. Another extreme example would be score distillation sampling (SDS) [128], where the guidance scale in the order of a few hundred is chosen. Using such a high guidance scale leads to better asset quality to some extent, but induces blurry and saturated results. Several research efforts have been made to mitigate this downside by exploring methods where using a smaller guidance scale suffices [173, 101]. Although recent progress in SDS-type methods has reduced the necessary guidance scale to a range that is similar to those of ancestral samplers, using a moderately large ω is considered an inevitable choice.

In this work, we aim to give an answer to this conundrum by revisiting the geometric view of diffusion models. In particular, inspired by the recent advances in diffusion-based inverse problem solvers (DIS) [78, 26, 152, 86, 27], we reformulate the text guidance as an inverse problem with a text-conditioned score-matching loss and derive a reverse diffusion sampling strategy by utilizing decomposed diffusion sampling (DDS) [27]. This results in a surprisingly simple fix of CFG to the sampling process without any computational overhead. The resulting process, which we call CFG++, works with a small guidance scale, typically $\lambda \in [0.0, 1.0]$, that smoothly *interpolates* between unconditional and conditional sampling, with $\lambda = 1.0$ having a similar effect as using CFG sampling with $\omega \sim 12.5$ at 50 neural function evaluation (NFE). Furthermore, DDIM inversion with CFG++ is invertible up to the discretization error, simplifying image editing. Comparing CFG++ against CFG shows that we achieve consistently better sample quality for text-to-image (T2I) generation, significantly better DDIM inversion capabilities that lead to enhanced reconstruction and editing, and enabling the incorporation of CFG guidance to diffusion inverse solvers (DIS) [26]. While the applications of CFG++ that we show in this work are limited, we believe that our work will have a broad impact that can be applied to all applications that leverage text guidance through the traditional CFG.

6.2.1 Background

Classifier free guidance For a conditional diffusion, [63] considered the sharpened posterior distribution $p^\omega(\boldsymbol{x}|\boldsymbol{c}) \propto p(\boldsymbol{x})p(\boldsymbol{c}|\boldsymbol{x})^\omega$. Using Bayes rule for some timestep t,

$$\nabla_{\boldsymbol{x}} \log p^\omega(\boldsymbol{x}_t|\boldsymbol{c}) = \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t) + \omega(\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{c}) - \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)) \tag{6.19}$$

Parametrizing the score function with $\boldsymbol{\epsilon}_\theta$, as in DDPM [62], we have

$$\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_t) := \hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t) + \omega[\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_t) - \hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t)] \tag{6.20}$$

where we introduce a compact notation $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}^{\omega}$ that guides the sampling from the sharpened posterior. When sampling with CFG guidance with DDIM sampling, one replaces $\hat{\boldsymbol{\epsilon}}_{\varnothing}$ with $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}^{\omega}$ for both the Tweedie estimate and the subsequent update step, leading to Algorithm 12.

Revisiting DIS Diffusion model-based inverse problem solvers (DIS) aims to perform posterior sampling from an unconditional diffusion model [78, 26, 152, 86]. Specifically, for a given loss function $\ell(\boldsymbol{x})$ which often stems from the likelihood, the goal of DIS is to address the optimization problem $\min_{\boldsymbol{x}\in\mathcal{M}} \ell(\boldsymbol{x})$, where $\mathcal{M}$ represents the clean data manifold sampled from the unconditional distribution $p_0(\boldsymbol{x})$. It is essential to navigate in a way that minimizes cost while also identifying the correct clean manifold.

[26] proposed diffusion posterior sampling (DPS), where the updated estimate from the noisy sample $\boldsymbol{x}_t \in \mathcal{M}_t$ is constrained to stay on the same noisy manifold $\mathcal{M}_t$. This is achieved by computing the manifold constrained gradient (MCG) [29] on a noisy sample $\boldsymbol{x}_t \in \mathcal{M}_t$ as $\nabla_{\boldsymbol{x}_t}^{mcg}\ell(\boldsymbol{x}_t) := \nabla_{\boldsymbol{x}_t}\ell(\hat{\boldsymbol{x}}_t)$, where $\hat{\boldsymbol{x}}_t$ is the denoised sample through Tweedie's formula [43]. The resulting algorithm with DDIM [156] can be stated as follows:

$$\boldsymbol{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\hat{\boldsymbol{x}}_{\varnothing} - \gamma_t \nabla_{\boldsymbol{x}_t}\ell(\hat{\boldsymbol{x}}_{\varnothing})\right) + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\boldsymbol{\epsilon}}_{\varnothing}, \tag{6.21}$$

where $\gamma_t > 0$ denotes the step size. Under the linear manifold assumption [29, 26], this allows precise transition to $\mathcal{M}_{t-1}$. To mitigate the computational complexity and the instability of neural network backprop, [27] shows that Eq. (6.21) can be equivalently represented as

$$\boldsymbol{x}_{t-1} \simeq \sqrt{\bar{\alpha}_{t-1}}\left(\hat{\boldsymbol{x}}_{\varnothing} - \gamma_t \nabla_{\hat{\boldsymbol{x}}_{\varnothing}}\ell(\hat{\boldsymbol{x}}_{\varnothing})\right) + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\boldsymbol{\epsilon}}_{\varnothing} \tag{6.22}$$

under further assumptions on $\mathcal{M}$. This method, often referred to as the decomposed diffusion sampling (DDS), bypasses the computation of the score Jacobian, similar to [128], making it stable and suitable for large-scale medical imaging inverse problems [27]. In the following, we leverage the insight from DDS to propose an algorithm to improve upon the CFG algorithm.

6.2.2 The CFG++ Algorithm

Derivation of the algorithm

Instead of uncritically adopting the sharpened posterior distribution $p^\omega(\boldsymbol{x}|\boldsymbol{c}) \propto p(\boldsymbol{x})p(\boldsymbol{c}|\boldsymbol{x})^\omega$ as introduced by [63], we adopt a fundamentally different strategy by reformulating text-guidance as an optimization problem. Specifically, our focus is on identifying a loss function $\ell(\boldsymbol{x})$ in Eq. (6.22) such that, when minimized under the condition set by the text, enables the reverse diffusion process to generate samples that increasingly satisfy the text condition progressively.

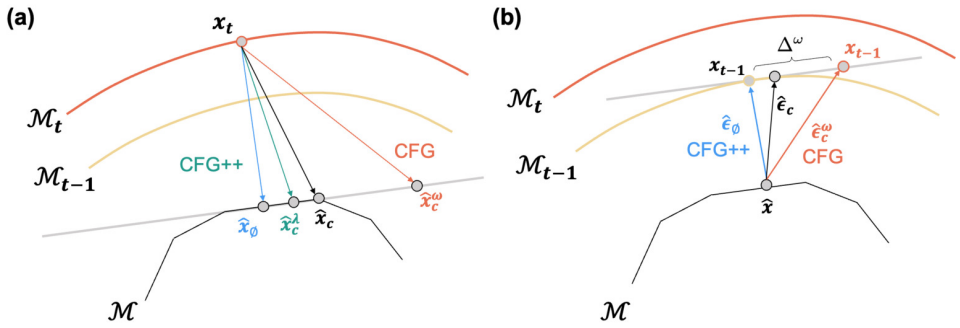


Figure 6.8: Off-manifold phenomenon of CFG arise from: (a) the typical CFG scale $\omega > 1.0$ which leads to extrapolation and deviation from the piecewise linear data manifold, and (b) CFG's renoising process, which introduces a nonzero offset Δ^ω from the correct manifold. CFG++ effectively mitigates all these artifacts.

One of the most significant contributions of this paper is to reveal that the text-conditioned score matching loss or alternatively, score distillation sampling (SDS) loss [128] is ideally suited for our purpose. Specifically, we are interested in solving the following inverse problem through diffusion models:

$$\min_{\boldsymbol{x}\in\mathcal{M}} \ell_{sds}(x), \quad \ell_{sds}(\boldsymbol{x}) := \|\boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\boldsymbol{x} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \boldsymbol{c}) - \boldsymbol{\epsilon}\|_2^2 \tag{6.23}$$

This implies that our goal is to identify solutions on the clean manifold $\mathcal{M}$ that optimally aligns with the text condition $\boldsymbol{c}$.

To avoid the Jacobian computation, in this paper, we attempt to solve Eq. (6.23) through DDS in Eq. (6.22). The resulting sampling process from reverse diffusion is then given by

$$\boldsymbol{x}_{t-1} \quad = \quad \sqrt{\bar{\alpha}_{t-1}}\left(\hat{\boldsymbol{x}}_\varnothing - \gamma_t \nabla_{\hat{\boldsymbol{x}}_\varnothing}\ell_{sds}(\hat{\boldsymbol{x}}_\varnothing)\right) + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\boldsymbol{\epsilon}}_\varnothing. \tag{6.24}$$

By using $\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$ from the clean image $\boldsymbol{x} \in \mathcal{M}$, we can equivalently write the loss as $\ell_{sds}(\boldsymbol{x}) = \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}\|\boldsymbol{x} - \hat{\boldsymbol{x}}_{\boldsymbol{c}}\|^2$, which leads to

$$\boldsymbol{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\hat{\boldsymbol{x}}_\varnothing + \lambda(\hat{\boldsymbol{x}}_{\boldsymbol{c}} - \hat{\boldsymbol{x}}_\varnothing)\right) + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\boldsymbol{\epsilon}}_\varnothing \tag{6.25}$$

where $\lambda := \frac{2\bar{\alpha}_t}{1-\bar{\alpha}_t}\gamma_t$. Using the CFG notation $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}^\lambda(\boldsymbol{x}_t) := \hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_t) + \lambda[\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_t) - \hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_t)]$ and $\hat{\boldsymbol{x}}_\varnothing + \lambda(\hat{\boldsymbol{x}}_{\boldsymbol{c}} - \hat{\boldsymbol{x}}_\varnothing) = (\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}^\lambda(\boldsymbol{x}_t))/\sqrt{\bar{\alpha}_t}$, Eq. (6.25) can be equivalently represented as

$$\hat{\boldsymbol{x}}_{\boldsymbol{c}}^\lambda(\boldsymbol{x}_t) = (\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}^\lambda(\boldsymbol{x}_t))/\sqrt{\bar{\alpha}_t} \tag{6.26}$$

$$\boldsymbol{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{x}}_{\boldsymbol{c}}^\lambda(\boldsymbol{x}_t) + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_t) \tag{6.27}$$

which is summarized in Algorithm 13. By examining Algorithm 12 and Algorithm 13, we observe that CFG and CFG++ are mostly the same, with a crucial difference in the renoising process. This surprisingly simple fix of utilizing the unconditional noise $\hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_t)$ instead of $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}^\omega(\boldsymbol{x}_t)$ leads to a smoother trajectory of generation, (Fig. 6.7 bottom) and generation with superior quality (Fig. 6.7 top).

Although we focus our construction of the solver on DDIM for simplicity, note that DDIM is just one way of reverse sampling. There are other widely-used solvers such as Karras Euler and its variants [79], DPM-solver [110, 111], their ancestral variants[7], etc. For most widely used solvers up to the second order, a single-step

[7] https://github.com/crowsonkb/k-diffusion

update of solving the unconditional PF-ODE can be represented as

$$\boldsymbol{x}_i = \hat{\boldsymbol{x}}_\varnothing(\boldsymbol{x}_{i-1}) + a_i\hat{\boldsymbol{x}}_\varnothing(\boldsymbol{x}_{i-1}) + b_i\hat{\boldsymbol{x}}_\varnothing(\boldsymbol{x}_{i-2}) + c_i\boldsymbol{x}_{i-1} + d_i\boldsymbol{\epsilon},\ \boldsymbol{\epsilon}\sim\mathcal{N}(0,\boldsymbol{I}), \tag{6.28}$$

with $d_i \neq 0$ when one uses an ancestral sampler. Note that the first right-hand side term in Eq. (6.28) corresponds to the denosing, whereas the rest terms describes the higher-order corrected version of the *renoising* process. As the goal of CFG++ is to optimize the denoising process under the text-guidance while keeping the renoising components equivalent to unconditional sampling, applying CFG++ to the general iteration in Eq. (6.28) simply leads to

$$\boldsymbol{x}_i = \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\lambda}(\boldsymbol{x}_{i-1}) + a_i\hat{\boldsymbol{x}}_\varnothing(\boldsymbol{x}_{i-1}) + b_i\hat{\boldsymbol{x}}_\varnothing(\boldsymbol{x}_{i-2}) + c_i\boldsymbol{x}_{i-1} + d_i\boldsymbol{\epsilon},\ \boldsymbol{\epsilon}\sim\mathcal{N}(0,\boldsymbol{I}). \tag{6.29}$$

Moreover, CFG++ naturally extends to distilled diffusion models, such as SDXL-turbo [142] and SDXL-lightning [104], where akin to Eq. (6.29), we use the conditional denoised estimate, but for the rest of the noise components in the Euler solver, we use the unconditional estimate. For details in how to apply CFG++ to various solvers, see Appendix A.3.

Geometry of CFG++

Mitigating off-manifold phenomenon In Fig. 6.7 (bottom), we illustrate the evolution of the posterior mean through Tweedie's formula during the reverse diffusion process. Notably, in the early phases of reverse diffusion sampling under CFG, there is a sudden shift in the image and intense color saturation. Conversely, CFG++ is free from the undesirable off-manifold phenomenon. In the following, we investigate why this is the case.

Note that the denoised estimate of CFG and CFG++ at time t can be equivalently represented as

$$\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\lambda}(\boldsymbol{x}_t) = (1-\lambda)\hat{\boldsymbol{x}}_\varnothing(\boldsymbol{x}_t) + \lambda\hat{\boldsymbol{x}}_{\boldsymbol{c}}(\boldsymbol{x}_t), \quad \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_t) = (1-\omega)\hat{\boldsymbol{x}}_\varnothing(\boldsymbol{x}_t) + \omega\hat{\boldsymbol{x}}_{\boldsymbol{c}}(\boldsymbol{x}_t). \tag{6.30}$$

$\lambda, \omega \in [0,1]$ facilitates an *interpolation*. However, with $\omega > 1.0$, CFG *extrapolates* beyond the unconditional and conditional estimates. Consequently, under the assumption that the clean manifold be piecewise linear [27], the conditional posterior mean estimates from CFG obtained with a guidance scale outside the range of $[0,1]$, can easily extend beyond the piecewise linear manifold. This may lead to the estimates potentially "falling off" the data manifold, as depicted by an orange arrow pointing downwards in Fig. 6.8(a). Thus, we select $\lambda \in [0,1]$ as the guidance scale for CFG++ to ensure it remains an *interpolation* between the unconditional and conditional estimate, thus preventing it from 'falling off' the clean data manifold. An additional source of the off-manifold phenomenon in CFG occurs during the transition from the clean manifold $\mathcal{M}$ to the subsequent noisy manifold $\mathcal{M}_{t-1}$, due to a similar off-manifold phenomenon from the large guidance scale (i.e. extrapolation), illustrated in Fig. 6.8 (b).

Text-Image alignment. In Fig. 6.7 (top), we observe enhanced text-to-image alignment achieved with CFG++. This enhanced alignment capability is a natural consequence of CFG++, which directly minimizes the text-conditioned score-matching loss as shown in Eq. (6.23). In contrast, CFG indirectly seeks text alignment through the sharpened posterior distribution $p^\omega(\boldsymbol{x}|\boldsymbol{c}) \propto p(\boldsymbol{x})p(\boldsymbol{c}|\boldsymbol{x})^\omega$. Therefore, CFG++ inherently outperforms CFG in terms of text alignment due to its fundamental design principle. For example, Fig. 6.9 displays the normalized text-conditioned score matching loss, represented as $(1-\bar{\alpha}_t)\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, \boldsymbol{c})\|^2/\bar{\alpha}_t = \|\boldsymbol{x} - \hat{\boldsymbol{x}}_{\boldsymbol{c}}\|^2$, throughout the reverse diffusion sampling process for both CFG and CFG++. The loss plot associated with CFG shows fluctuations and maintains a noticeable gap compared to CFG++ even after the completion of the reverse diffusion process. In fact, the fluctuation associated with CFG is also related to the off-manifold issue, and from Fig. 6.9 we can easily see that the off-manifold phenomenon is more dominant at early stage of reverse diffusion sampling. Conversely, the loss

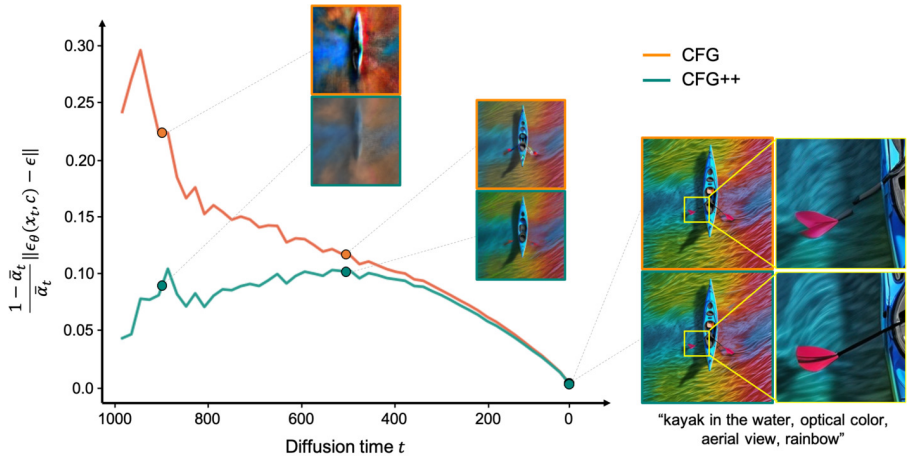


Figure 6.9: Text-conditioned score matching loss throughout the reverse diffusion sampling for both CFG and CFG++ in SDXL. Avg. loss computed with 55 prompts from [20].

trajectory for CFG++ demonstrates a much smoother variation, particularly during the early stages of reverse diffusion.

DDIM inversion. As discussed in [151], the denoising process for unconditional DDIM is approximately invertible, meaning that $\boldsymbol{x}_t$ can generally be recovered from $\boldsymbol{x}_{t-1}$ Specifically, from the DDIM update steps, we have the following approximate inversion formula for unconditional DDIM:

$$\hat{\boldsymbol{x}}_\varnothing(\boldsymbol{x}_t) = (\boldsymbol{x}_{t-1} - \sqrt{1-\bar{\alpha}_{t-1}}\hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_t))/\sqrt{\bar{\alpha}_{t-1}} \simeq (\boldsymbol{x}_{t-1} - \sqrt{1-\bar{\alpha}_{t-1}}\hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_{t-1}))/\sqrt{\bar{\alpha}_{t-1}} \tag{6.31}$$

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\hat{\boldsymbol{x}}_\varnothing(\boldsymbol{x}_t) + \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_t) \simeq \sqrt{\bar{\alpha}_t}\hat{\boldsymbol{x}}_\varnothing(\boldsymbol{x}_t) + \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_{t-1}) \tag{6.32}$$

where the approximation arises from $\hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_t) \simeq \hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_{t-1})$. A similar inversion procedure has been employed for conditional DDIM inversion under CFG by assuming $\hat{\boldsymbol{\epsilon}}^\omega_{\boldsymbol{c}}(\boldsymbol{x}_t) \simeq \hat{\boldsymbol{\epsilon}}^\omega_{\boldsymbol{c}}(\boldsymbol{x}_{t-1})$, and replacing $\hat{\boldsymbol{\epsilon}}_\varnothing$ by $\hat{\boldsymbol{\epsilon}}^\omega_{\boldsymbol{c}}$ as detailed in Algorithm 14.

On the other hand, by examining Eq. (6.26) and Eq. (6.27), we can obtain the following approximate DDIM inversion formula under CFG++:

$$\hat{\boldsymbol{x}}^\lambda_{\boldsymbol{c}}(\boldsymbol{x}_t) \simeq (\boldsymbol{x}_{t-1} - \sqrt{1-\bar{\alpha}_{t-1}}\hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_{t-1}))/\sqrt{\bar{\alpha}_{t-1}} \tag{6.33}$$

$$\boldsymbol{x}_t \simeq \sqrt{\bar{\alpha}_t}\hat{\boldsymbol{x}}^\lambda_{\boldsymbol{c}}(\boldsymbol{x}_t) + \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}^\lambda_{\boldsymbol{c}}(\boldsymbol{x}_{t-1}) \tag{6.34}$$

where we apply the approximation $\hat{\boldsymbol{\epsilon}}^\lambda_{\boldsymbol{c}}(\boldsymbol{x}_t) \simeq \hat{\boldsymbol{\epsilon}}^\lambda_{\boldsymbol{c}}(\boldsymbol{x}_{t-1})$ alongside the usual assumption $\hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_t) \simeq \hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_{t-1})$. This formulation underpins the CFG++ guided DDIM inversion algorithm, presented in Algorithm 15.

In practice, for small time step size the error $\hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_t) \simeq \hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_{t-1})$ is relatively small, so the unconditional DDIM inversion by Eq. (6.31) and Eq. (6.32) lead to relatively insignificant errors. Unfortunately, the corresponding inversion from conditional diffusion under CFG is quite distorted as noted in [116, 166]. In fact, this distortion is originated from the inaccuracy of the approximation $\hat{\boldsymbol{\epsilon}}^\omega_{\boldsymbol{c}}(\boldsymbol{x}_t) \simeq \hat{\boldsymbol{\epsilon}}^\omega_{\boldsymbol{c}}(\boldsymbol{x}_{t-1})$. More specifically, even when $\hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_t) \simeq \hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_{t-1})$, the approximation error from the CFG is given by

$$\begin{aligned}\boldsymbol{\varepsilon}_{cfg} &:= \hat{\boldsymbol{\epsilon}}^\omega_{\boldsymbol{c}}(\boldsymbol{x}_t) - \hat{\boldsymbol{\epsilon}}^\omega_{\boldsymbol{c}}(\boldsymbol{x}_{t-1}) = (\hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_t) - \hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_{t-1})) + \omega\big(\delta\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_t) - \delta\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_{t-1})\big) \\ &\simeq \omega\big(\delta\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_t) - \delta\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_{t-1})\big),\end{aligned} \tag{6.35}$$

where $\delta\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_t) := \hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_t) - \hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_t)$ denotes the directional component from unconditional to the conditional diffusion. Since the directional component by the text guidance is not negligible, this error becomes significant for

	$\omega = 2.0, \lambda = 0.2$		$\omega = 5.0, \lambda = 0.4$		$\omega = 7.5, \lambda = 0.6$		$\omega = 9.0, \lambda = 0.8$		$\omega = 12.5, \lambda = 1.0$	
Method	FID ↓	CLIP ↑	FID ↓	CLIP ↓	FID ↓	CLIP ↑	FID ↓	CLIP ↑	FID ↓	CLIP ↑
CFG [63]	13.84	0.298	15.08	0.310	17.71	0.312	20.01	0.312	21.23	0.313
CFG++ (ours)	**12.75**	**0.303**	**14.95**	0.310	**17.47**	0.312	**19.34**	**0.313**	**20.88**	0.313

Table 6.6: Quantitative evaluation of 50NFE DDIM T2I with SD v1.5 on COCO 10k

high guidance scale ω. Accordingly, the guidance scale must be heavily downweighted in order for inversions on real world images to be stable, thus limiting the strength of edits. To mitigate this issue, the authors in [116, 166] developed null text optimization and coupled transform techniques, respectively.

On the other hand, under the usual DDIM assumption $\hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_t) \simeq \hat{\boldsymbol{\epsilon}}_\varnothing(\boldsymbol{x}_{t-1})$, the approximation error of CFG++ mainly arises from Eq. (6.34), which is smaller than that of CFG since we have

$$\|\boldsymbol{\varepsilon}_{cfg++}\| = \lambda \|\delta\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_t) - \delta\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_{t-1})\| < \|\boldsymbol{\varepsilon}_{cfg}\| \tag{6.36}$$

thanks to $\lambda < \omega$. Therefore, CFG++ significantly improves the DDIM inversion as shown Fig. 6.7 (middle) for representative results and Sec. 6.2.4 for further discussions.

6.2.3 Experimental Results

In this section, we design experiments to show the limitations of CFG and how CFG++ can effectively mitigate these downsides. The main experiments were conducted with SD v1.5 or SDXL with 50 NFE DDIM sampling. In this regime, we searched for the matching guidance values of ω and λ for a fair comparison. We fix $\lambda = 0.2, 0.4, 0.6, 0.8, 1.0$ and find the ω values that produce the images that are of closest proximity in terms of LPIPS distance given the same seed. We found that the corresponding values were $\omega = 2.0, 5.0, 7.5, 9.0, 12.5$, respectively. Some of the experiments were also conducted with distilled model such as SDXL-turbo, lightning.

Text-to-Image Generation

Using the corresponding scales for ω and λ, we directly compare the performance of the T2I task using SD v1.5 and SDXL. In Tab. 6.6, we report quantitative metrics using 10k images generated from COCO captions [105]. Here, we observe a constant improvement of the FID metric across all guidance scales, with approximately the same level of CLIP similarity or better. The improvements can also be clearly seen in Fig. 6.10 (SD v1.5), where the unnatural components of the generated images are corrected. Specifically, we see that unnatural depictions of human hands, and incorrect renderings of the text are corrected in CFG++, a long-standing research question in and of its own [127, 125, 19]. We find that the improvement gain from CFG++ is even more dramatic for distilled diffusion models such as SDXL-{turbo, lightning}. We quantify the results on the same 5k prompts with 6 NFE sampling. Here, we see significant boosts in the quality of the generated images, which is depicted in the improvements seen in Tab. 6.7.

Model	Metric	CFG	CFG++ (ours)
SDXL-Turbo	FID↓	59.67	**59.21**
	CLIP↑	0.320	**0.325**
	ImageReward↑	0.777	**0.968**
SDXL-Lightning	FID↓	56.11	**55.19**
	CLIP↑	0.322	**0.324**
	ImageReward↑	0.691	**0.829**
SD v1.5 (DPM++ 2M)	FID↓	32.72	**32.58**
	CLIP↑	**0.313**	0.312

Table 6.7: Quant. eval. on accelerated T2I sampling

To show the compatibility of CFG++ with higher-order solvers, we experiment with DPM++2M [111] solver with 20 NFE. We note that in such low NFE regime, a CFG scale of 5.0 corresponds to the CFG++ scale of 1.0, and

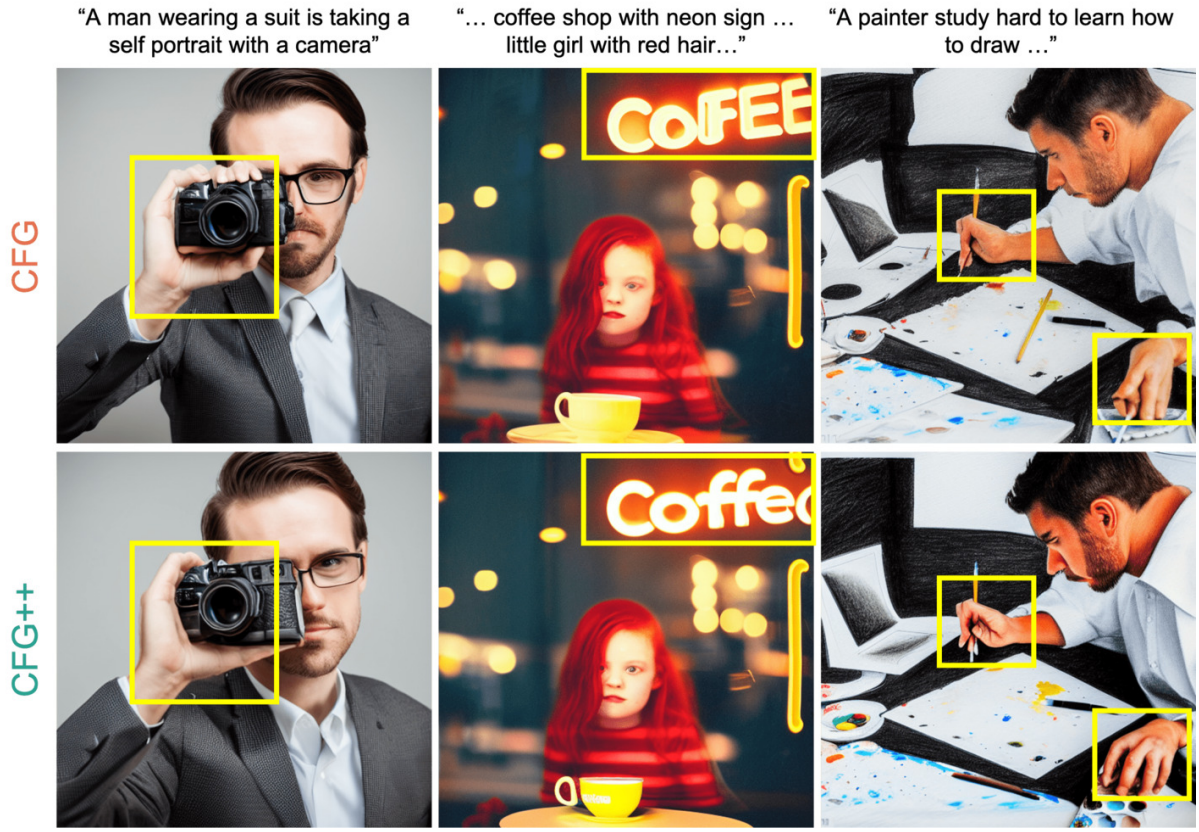


Figure 6.10: T2I using SD v1.5, CFG vs CFG++ ($\omega = 9.0, \lambda = 0.8$). Unnatural depictions of human hands, and incorrect renderings of the text by CFG are corrected in CFG++.

Figure 6.11: T2I using SDXL-{turbo, lightning}, 6 NFE, CFG vs CFG++.

we have to slightly extrapolate the value of $\lambda \geq 1.0$ to achieve stronger guidance results. In Tab. 6.7, we again see that CFG++ yields results that are favorable to standard CFG.

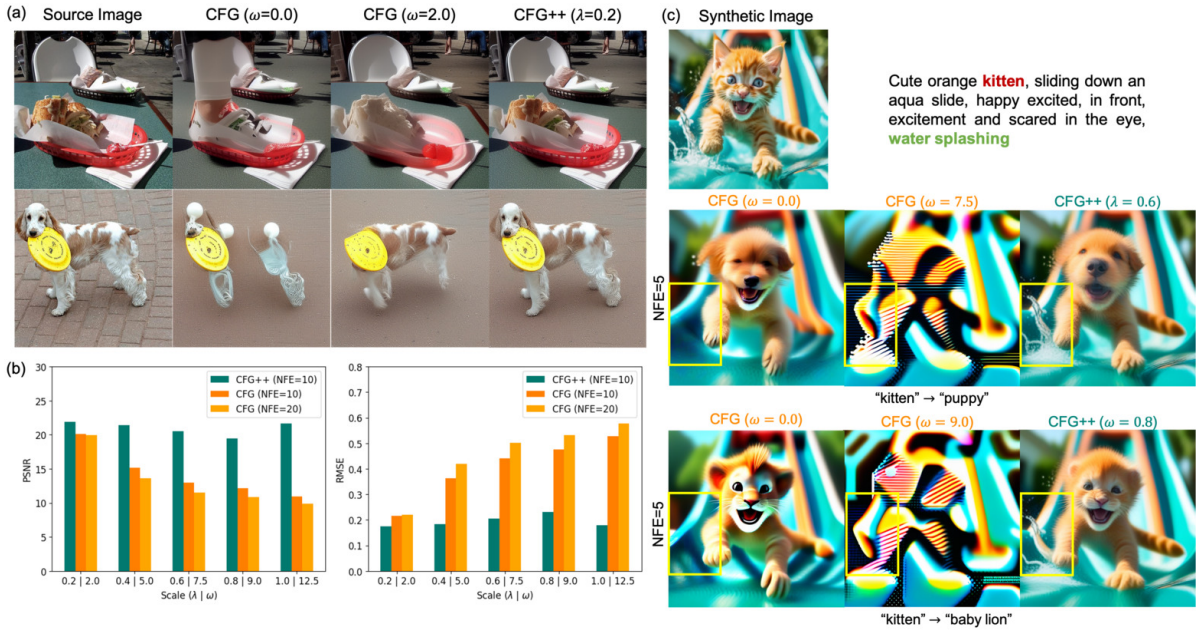


Figure 6.12: Inversion and editing results. (a) Reconstructed samples after inversion by CFG and CFG++. (b) Quantitative comparison between CFG and CFG++ for reconstruction. (c) Image editing comparison via SDXL.

6.2.4 Diffusion image Inversion and Editing

We further explore the effect of CFG++ on DDIM inversion [39], where the source image is reverted to a latent vector that can reproduce the original image through generative process. DDIM inversion is well-known to break down in the usual CFG setting as CFG magnifies the accumulated error of each inversion step [116], violating local linearization assumptions [166]. We show that CFG++ mitigates this issue by improving the inversion and image editing capabilities. We evaluate our method following the experimental setups in [124] and [86].

Diffusion Image Inversion. Using the matched set of scales for ω and λ, we demonstrate the effect of CFG++ on the diffusion image inversion task. Specifically, we reconstruct the images after inversion and evaluate it through PSNR and RMSE, following the methodology of [166]. In Fig. 6.12, we illustrate the reconstructed examples (6.12a) from a real image from COCO data set and computed metrics (6.12b) for 5k COCO-2014 [105] validation set. Both qualitative and quantitative evaluations demonstrate a consistent improvement on reconstruction performance induced by CFG++. Notably, DDIM inversion with CFG++ leads to a consistent reconstruction of the source image across all guidance scales, while DDIM inversion with CFG fails to reconstruct it.

Image Editing. Fig. 6.12c compares the image editing result using CFG and CFG++ followed by image inversion with 5 NFEs. In the image editing stage, a word in the source text highlighted by green color is swapped with the target concept, and this modified text is used as the condition for sampling. CFG++ successfully edits the target concept while preserving other concepts, such as background. In particular, the water splashing in the background, which tends to disappear in the conventional CFG, is maintained through the inversion process by CFG++. This emphasizes CFG++'s superior ability to retain specific scene elements that are frequently lost in the standard CFG approach. Moreover, standard CFG's disrupted DDIM inversion leads to saturated and less faithful editing results, whereas CFG++ delivers precise and high-fidelity edits.

Text-Conditioned Inverse Problems

Inverse problem involves restoring the original data $\boldsymbol{x}$ from a noisy measurement $\boldsymbol{y} = \mathcal{A}(\boldsymbol{x}) + \boldsymbol{n}$, where $\mathcal{A}$ represents an imaging operator introduces distortions (e.g, Gaussian blur) and $\boldsymbol{n}$ denotes the measurement noise. Diffusion inverse solvers (DIS) address this challenge by leveraging pre-trained diffusion models as implicit

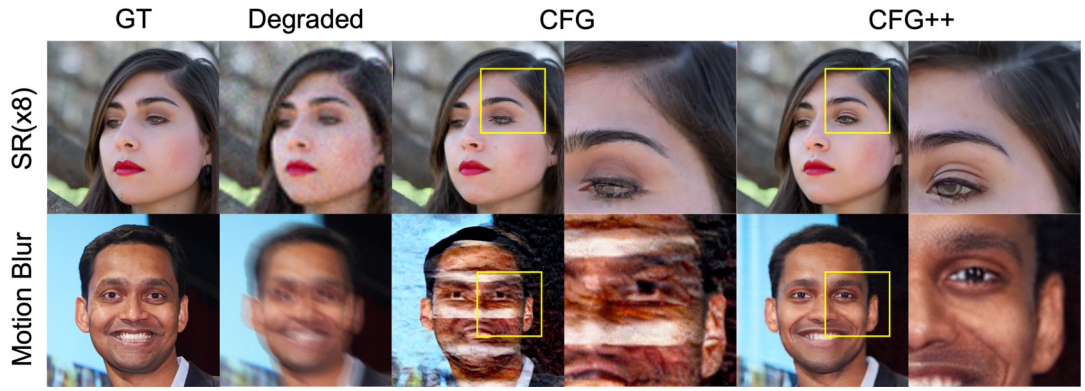


Figure 6.13: Qualitative comparison on various inverse problems using PSLD [138] under CFG and CFG++.

priors and performing posterior sampling $\boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{y})$. Methods that leverage latent diffusion have gained recent interest [138, 150], but leveraging texts for solving these problems remains relatively underexplored [32, 85]. One of the main reasons for this is that naively using CFG on top of latent DIS leads to diverging samples [32]. Several heuristic modifications such as null prompt optimization [85] with modified sampling schemes were needed to mitigate this drawback. This naturally leads to the question: Is it possible to leverage CFG guidance as a plug-and-play component of existing solvers? Here, we answer this question with a positive by showing that CFG++ enables the incorporation of text prompts into a standard solver. Specifically, we focus on comparing the performance of PSLD [138] combined with CFG and CFG++ in solving linear inverse problems. This evaluation utilizes the FFHQ [80] 512x512 dataset and the text prompt "a high-quality photo of a face".

As shown in Tab. 6.8, our method mostly outperforms both the vanilla PSLD and PSLD with CFG. The superiority is also evident from the Fig. 6.13, where CFG++ consistently delivers high-quality reconstructions across all tasks. PSLD with CFG often suffer from artifacts and blurriness. Conversely, CFG++ achieves better fidelity, clearly distinguishing between faces and faithfully reproducing fine details like eyelids and hair texture.

	SR (x8)			Deblur (motion)			Deblur (gauss)			Inpaint		
Method	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑
PSLD	46.24	0.413	24.41	97.51	0.500	21.83	41.65	**0.388**	26.88	10.27	9.36	30.15
PSLD + CFG	41.24	0.394	**24.91**	91.90	0.493	**22.29**	41.52	0.390	**26.94**	**9.36**	0.055	30.27
PSLD + CFG++ (ours)	**36.58**	**0.385**	24.87	**65.67**	**0.482**	21.93	**39.85**	0.400	26.90	9.78	**0.052**	**30.31**

Table 6.8: Quantitative comparison (FID, LPIPS, PSNR) of PSLD, PSLD with CFG, and PSLD with CFG++ on Latent Diffusion Inverse Solver.

6.2.5 Related Works and Discussions

[96] observed that applying CFG guidance at the earlier stages of sampling always led to detrimental effects and drastically reduced the diversity of the samples. The guidance at the later stages of sampling had minimal effects. Drawing upon these observations, it was empirically shown that applying CFG only in the limited time interval near the middle led to the best performance. Similar observations were made in the SD community [65, 8] in adjusting the guidance scale across t. These findings are orthogonal to ours in that these methods keep the sampling trajectory the same and try to empirically adjust the strength of the guidance, while we aim to design a different trajectory. More recently, [12] observed that the CFG score is not a valid denoising direction, showing that in the asymptotic limit, CFG sampling can be considered as a specific type of predictor-corrector (PC) sampling, dubbed PCG. PCG shares a similar spirit with CFG++ in that for the mainstream of sampling, one defers from the use of mixed CFG score (we use unconditional score, PCG uses conditional score) to avoid invalid directions. While PCG derives an interesting connection of PC sampling with CFG, the proposed algorithm does not improve

the sampling performance. In contrast, CFG++ improves the sample quality without additional computation overhead. Since CFG++ attempts to make adjustments of CFG using a linear combination of score functions, it is not surprising to see that CFG++ sampling can be achieved by setting a time varying schedule. Specifically, by setting a time-varying schedule ω_t as $\omega_t = -\gamma_t/\xi_t$, where $\gamma_t = \frac{\sqrt{\bar{\alpha}_{t-1}}\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}, \xi_t = \sqrt{1-\bar{\alpha}_{t-1}} - \frac{\sqrt{\bar{\alpha}_{t-1}}\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}$, one can achieve the same effect with CFG as CFG++. However, this specific choice of time-varying guidance scale has not been reported in the literature, as it is a schedule that is relatively complex to be drawn heuristically. We further show in Appendix A.4 that such choice removes undesirable oscillatory behavior in the evolution of the posterior mean.

Chapter 7. Conclusion

In this thesis, we developed methods for solving inverse problems with diffusion models in a plug-and-play fashion, eliminating the need for additional training. All methods presented aimed to enable practical approximations of posterior sampling. Unlike conventional reconstructions derived as point estimates through supervised learning, posterior samples obtained through DIS exhibit superior perceptual quality by more closely aligning with the true data distribution. The ability to obtain multiple posterior samples allows for estimating the approximate MMSE by averaging over samples, providing results comparable to those from supervised training. Additionally, the pixel-wise variance across samples offers a straightforward path to uncertainty estimation.

Throughout the chapters, we emphasized the strengths of each work. Here, we discuss some limitations and areas for future improvement.

1. Much of this thesis focused on (1) broadening the applicability of DIS—from linear to non-linear and from non-blind to blind settings, addressing prior mismatch, etc.—and (2) aiming for practical deployment, with efforts toward accelerating diffusion sampling and tailoring it to medical imaging. While we presented theoretical insights to clarify the approximation error each method introduces, we placed less emphasis on achieving exact posterior sampling—a direction of growing interest [57]. Extending DIS for broader deployment in scientific imaging and beyond may benefit from reducing the gap between practical solvers and exact posterior samplers, potentially improving theoretical guarantees alongside performance.

2. Inverse problems are pervasive beyond imaging. They arise, for example, in estimating parameters of PDEs [71] or inferring camera positions from sparse image sets [168]. While DIS shows promise, applying it to these diverse applications requires further adaptation and practical considerations, especially to generalize across these varied problem settings.

3. DIS is inherently iterative. Even methods considered fast require several tens of NFEs to achieve quality results, which may still be too slow for time-sensitive applications. Recent advances in diffusion model distillation [153] offer promising pathways to accelerate inference, and incorporating these methods could help DIS meet the demands of time-critical tasks.

Interest in this area has grown rapidly, and although progress is significant, each solution raises new questions. As we continue to refine and expand these methods, it's worth recalling that science is a journey of ever-deeper exploration. To conclude, I find the following quote by Earl C. Kelley encapsulates this dynamic process of discovery:

> *"We have not succeeded in answering all our problems. The answers we have found only serve to raise a whole set of new questions. In some ways we feel we are as confused as ever, but we believe we are confused on a higher level and about more important things."*—Earl C. Kelley

Bibliography

[1] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018.

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

[3] Daniel Otero Baguer, Johannes Leuschner, and Maximilian Schmidt. Computed tomography reconstruction using deep image prior and learned reconstruction methods. *Inverse Problems*, 36(9):094004, 2020.

[4] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. 2023.

[5] Riccardo Barbano, Alexander Denker, Hyungjin Chung, Tae Hoon Roh, Simon Arrdige, Peter Maass, Bangti Jin, and Jong Chul Ye. Steerable conditional diffusion for out-of-distribution adaptation in imaging inverse problems. *arXiv preprint arXiv:2308.14409*, 2023.

[6] Riccardo Barbano, Johannes Leuschner, Maximilian Schmidt, Alexander Denker, Andreas Hauptmann, Peter Maass, and Bangti Jin. An educated warm start for deep image prior-based micro ct reconstruction. *IEEE Transactions on Computational Imaging*, 8:1210–1222, 2022.

[7] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[8] Alex Birch. Turning off classifier-free guidance at low noise levels. https://twitter.com/Birchlabs/status/1640033271512702977, 2023. Idea mentioned on Twitter.

[9] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6228–6237, 2018.

[10] Kai Tobias Block, Martin Uecker, and Jens Frahm. Undersampled radial MRI with multiple coils. Iterative image reconstruction using a total variation constraint. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 57(6):1086–1098, 2007.

[11] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[12] Arwen Bradley and Preetum Nakkiran. Classifier-free guidance is a predictor-corrector. *arXiv preprint arXiv:2408.09000*, 2024.

[13] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*, 2019.

[14] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[15] Wai Ho Chak, Chun Pong Lau, and Lok Ming Lui. Subsampled turbulence removal network. *arXiv preprint arXiv:1807.04418*, 2018.

[16] Stanley H Chan. Tilt-then-blur or blur-then-tilt? clarifying the atmospheric turbulence model. *IEEE Signal Processing Letters*, 29:1833–1837, 2022.

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

[18] Guangyong Chen, Fengyuan Zhu, and Pheng Ann Heng. An efficient statistical method for image noise level estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 477–485, 2015.

[19] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser: Diffusion models as text painters. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[20] Junsong Chen, Jincheng YU, Chongjian GE, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-a: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

[21] Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. *arXiv preprint arXiv:2302.07194*, 2023.

[22] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.

[23] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. ILVR: Conditioning method for denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[24] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020.

[25] Hyungjin Chung, Jeongsol Kim, Sehui Kim, and Jong Chul Ye. Parallel diffusion models of operator and image for blind inverse problems. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[26] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

[27] Hyungjin Chung, Suhyeon Lee, and Jong Chul Ye. Decomposed diffusion sampler for accelerating large-scale inverse problems. In *International Conference on Learning Representations*, 2024.

[28] Hyungjin Chung, Dohoon Ryu, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Solving 3d inverse problems using pre-trained 2d diffusion models. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[29] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[30] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-Closer-Diffuse-Faster: Accelerating Conditional Diffusion Models for Inverse Problems through Stochastic Contraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[31] Hyungjin Chung and Jong Chul Ye. Score-based diffusion models for accelerated mri. *Medical Image Analysis*, page 102479, 2022.

[32] Hyungjin Chung, Jong Chul Ye, Peyman Milanfar, and Mauricio Delbracio. Prompt-tuning latent diffusion models for inverse problems. *arXiv preprint arXiv:2310.01110*, 2023.

[33] Event Horizon Telescope Collaboration et al. First m87 event horizon telescope results. iv. imaging the central supermassive black hole. *arXiv preprint arXiv:1906.11241*, 2019.

[34] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020.

[35] Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. *arXiv preprint arXiv:2401.11605*, 2024.

[36] Mauricio Delbracio and Peyman Milanfar. Inversion by direct iteration: An alternative to denoising diffusion for image restoration. *arXiv preprint arXiv:2303.11435*, 2023.

[37] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[38] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012.

[39] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[40] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[41] Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International conference on machine learning*. PMLR, 2023.

[42] Nicholas Dwork, Corey A Baron, Ethan MI Johnson, Daniel O'Connor, John M Pauly, and Peder EZ Larson. Fast variable density poisson-disc sample generation with directional variation for compressed sensing in mri. *Magnetic Resonance Imaging*, 77:186–193, 2021.

[43] Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

[44] Matthias J Ehrhardt and Marta M Betcke. Multicontrast mri reconstruction with structure-guided total variation. *SIAM Journal on Imaging Sciences*, 9(3):1084–1106, 2016.

[45] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024.

[46] Zalan Fabian, Reinhard Heckel, and Mahdi Soltanolkotabi. Data augmentation for deep learning based accelerated mri reconstruction with limited data. In *International Conference on Machine Learning*, pages 3057–3067. PMLR, 2021.

[47] Berthy T Feng, Jamie Smith, Michael Rubinstein, Huiwen Chang, Katherine L Bouman, and William T Freeman. Score-based diffusion models as principled priors for inverse imaging. *arXiv preprint arXiv:2304.11751*, 2023.

[48] C Fienup and J Dainty. Phase retrieval and image reconstruction for astronomy. *Image recovery: theory and application*, 231:275, 1987.

[49] James R Fienup. Reconstruction of an object from the modulus of its fourier transform. *Optics letters*, 3(1):27–29, 1978.

[50] James R Fienup. Phase retrieval algorithms: a comparison. *Applied optics*, 21(15):2758–2769, 1982.

[51] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[52] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

[53] Xiang Gao, Meera Sitharam, and Adrian E Roitberg. Bounds on the jensen gap, and implications for mean-concentrated distributions. *arXiv preprint arXiv:1712.05267*, 2017.

[54] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22930–22941, 2023.

[55] Mario González, Andrés Almansa, and Pauline Tan. Solving inverse problems by joint posterior maximization with autoencoding prior. *SIAM Journal on Imaging Sciences*, 15(2):822–859, 2022.

[56] Harshit Gupta, Michael T McCann, Laurene Donati, and Michael Unser. Cryogan: A new reconstruction paradigm for single-particle cryo-em via deep adversarial learning. *IEEE Transactions on Computational Imaging*, 7:759–774, 2021.

[57] Shivam Gupta, Ajil Jalal, Aditya Parulekar, Eric Price, and Zhiyang Xun. Diffusion posterior sampling is computationally intractable. *arXiv preprint arXiv:2402.12727*, 2024.

[58] Linchao He, Hongyu Yan, Mengting Luo, Kunming Luo, Wang Wang, Wenchao Du, Hu Chen, Hongyu Yang, and Yi Zhang. Iterative reconstruction based on latent diffusion model for sparse data reconstruction. *arXiv preprint arXiv:2307.12070*, 2023.

[59] Reinhard Heckel and Paul Hand. Deep decoder: Concise image representations from untrained non-convolutional networks. *arXiv preprint arXiv:1810.03982*, 2018.

[60] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

[61] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

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

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

[64] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pages 13213–13232. PMLR, 2023.

[65] Jeremy Howard and Rekil Prashanth. Adjusting guidance weight as a function of time. https://twitter.com/jeremyphoward/status/1584771100378288129, 2022. Idea mentioned on Twitter.

[66] Stephan Hoyer, Jascha Sohl-Dickstein, and Sam Greydanus. Neural reparameterization improves structural optimization. *arXiv preprint arXiv:1909.04240*, 2019.

[67] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[68] Zhe Hu and Ming-Hsuan Yang. Good regions to deblur. In *European conference on computer vision*, pages 59–72. Springer, 2012.

[69] Chin-Wei Huang, Jae Hyun Lim, and Aaron Courville. A variational perspective on diffusion-based generative models and score matching. *arXiv preprint arXiv:2106.02808*, 2021.

[70] Ajil Jalal, Marius Arvinte, Giannis Daras, Eric Price, Alexandros G Dimakis, and Jonathan Tamir. Robust compressed sensing mri with deep generative priors. *Advances in Neural Information Processing Systems*, 34, 2021.

[71] Enze Jiang, Jishen Peng, Zheng Ma, and Xiong-Bin Yan. Ode-dps: Ode-based diffusion posterior sampling for inverse problems in partial differential equation. *arXiv preprint arXiv:2404.13496*, 2024.

[72] Álvaro Barbero Jiménez. Mixture of diffusers for scene composition and high resolution image generation. *arXiv preprint arXiv:2302.02412*, 2023.

[73] Darui Jin, Ying Chen, Yi Lu, Junzhang Chen, Peng Wang, Zichao Liu, Sheng Guo, and Xiangzhi Bai. Neutralizing the impact of atmospheric turbulence on complex scene imaging via deep learning. *Nature Machine Intelligence*, 3(10):876–884, 2021.

[74] Kyong Hwan Jin, Michael T McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017.

[75] Yeonsik Jo, Se Young Chun, and Jonghyun Choi. Rethinking deep image prior for denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5087–5096, 2021.

[76] Hong Jung, Kyunghyun Sung, Krishna S Nayak, Eung Yeop Kim, and Jong Chul Ye. k-t focuss: a general compressed sensing framework for high resolution dynamic mri. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 61(1):103–116, 2009.

[77] Zahra Kadkhodaie and Eero Simoncelli. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. In *Advances in Neural Information Processing Systems*, volume 34, pages 13242–13254. Curran Associates, Inc., 2021.

[78] Zahra Kadkhodaie and Eero P Simoncelli. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

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

[80] 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.

[81] Masaki Katsura, Izuru Matsuda, Masaaki Akahane, Jiro Sato, Hiroyuki Akai, Koichiro Yasaka, Akira Kunimatsu, and Kuni Ohtomo. Model-based iterative reconstruction technique for radiation dose reduction in chest ct: comparison with the adaptive statistical iterative reconstruction technique. *European radiology*, 22(8):1613–1623, 2012.

[82] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

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

[84] Dongjun Kim, Seungjae Shin, Kyungwoo Song, Wanmo Kang, and Il-Chul Moon. Soft truncation: A universal training technique of score-based diffusion model for high precision score estimation. *International conference on machine learning*, 2022.

[85] Jeongsol Kim, Geon Yeong Park, Hyungjin Chung, and Jong Chul Ye. Regularization by texts for latent diffusion inverse solvers. *arXiv preprint arXiv:2311.15658*, 2023.

[86] Jeongsol Kim, Geon Yeong Park, and Jong Chul Ye. Dreamsampler: Unifying diffusion sampling and score distillation for image manipulation. *arXiv preprint arXiv:2403.11415*, 2024.

[87] Kwanyoung Kim and Jong Chul Ye. Noise2Score: Tweedie's Approach to Self-Supervised Image Denoising without Clean Images. *Advances in Neural Information Processing Systems*, 34, 2021.

[88] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[89] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[90] Robert Hildreth Kingston. *Detection of optical and infrared radiation*, volume 10. Springer, 2013.

[91] Dana A Knoll and David E Keyes. Jacobian-free newton–krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, 2004.

[92] Jan J Koenderink. The structure of images. *Biological cybernetics*, 50(5):363–370, 1984.

[93] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[94] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. *Advances in neural information processing systems*, 22, 2009.

[95] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8878–8887, 2019.

[96] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *arXiv preprint arXiv:2404.07724*, 2024.

[97] Anish Lahiri, Marc Klasky, Jeffrey A Fessler, and Saiprasad Ravishankar. Sparse-view cone beam ct reconstruction using data-consistent supervised and adversarial learning from scarce training data. *arXiv preprint arXiv:2201.09318*, 2022.

[98] Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338, 2000.

[99] 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.

[100] Suhyeon Lee, Hyungjin Chung, Minyoung Park, Jonghyuk Park, Wi-Sun Ryu, and Jong Chul Ye. Improving 3d imaging with pre-trained perpendicular 2d diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10710–10720, 2023.

[101] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. *arXiv preprint arXiv:2311.11284*, 2023.

[102] Jörg Liesen and Zdenek Strakos. *Krylov subspace methods: principles and analysis*. Oxford University Press, 2013.

[103] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5404–5411, 2024.

[104] Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024.

[105] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th*

European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pages 740–755. Springer, 2014.

[106] Tony Lindeberg. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2):225–270, 1994.

[107] Jiaming Liu, Yu Sun, Xiaojian Xu, and Ulugbek S Kamilov. Image restoration using total variation regularized deep image prior. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7715–7719. Ieee, 2019.

[108] Lu Liu. Model-based iterative reconstruction: a promising algorithm for today's computed tomography imaging. *Journal of Medical imaging and Radiation sciences*, 45(2):131–136, 2014.

[109] Yan Liu, Jianhua Ma, Yi Fan, and Zhengrong Liang. Adaptive-weighted total variation minimization for sparse data toward low-dose x-ray computed tomography image reconstruction. *Physics in Medicine & Biology*, 57(23):7923, 2012.

[110] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems*, 2022.

[111] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022.

[112] Michael Lustig, David Donoho, and John M Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

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

[114] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2014.

[115] Taylor R Moen, Baiyu Chen, David R Holmes III, Xinhui Duan, Zhicong Yu, Lifeng Yu, Shuai Leng, Joel G Fletcher, and Cynthia H McCollough. Low-dose ct image and projection dataset. *Medical physics*, 48(2):902–911, 2021.

[116] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023.

[117] Naoki Murata, Koichi Saito, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Yuki Mitsufuji, and Stefano Ermon. Gibbsddrm: A partially collapsed gibbs sampler for solving blind inverse problems with denoising diffusion restoration. In *International conference on machine learning*. PMLR, 2023.

[118] Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.

[119] Gregory Ongie, Ajil Jalal, Christopher A Metzler, Richard G Baraniuk, Alexandros G Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020.

[120] Jinshan Pan, Zhe Hu, Zhixun Su, and Ming-Hsuan Yang. l_0-regularized intensity and gradient prior for deblurring text images and beyond. *IEEE transactions on pattern analysis and machine intelligence*, 39(2):342–355, 2016.

[121] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1628–1636, 2016.

[122] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Deblurring images via dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2315–2328, 2017.

[123] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

[124] Geon Yeong Park, Jeongsol Kim, Beomsu Kim, Sang Wan Lee, and Jong Chul Ye. Energy-based cross attention for bayesian context update in text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

[125] Anton Pelykh, Ozge Mercanoglu Sincan, and Richard Bowden. Giving a hand to diffusion models: a two-stage approach to improving conditional human image generation. *arXiv preprint arXiv:2403.10731*, 2024.

[126] Daniele Perrone and Paolo Favaro. Total variation blind deconvolution: The devil is in the details. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2909–2916, 2014.

[127] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[128] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022.

[129] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2023.

[130] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[131] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[132] Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, and Wangmeng Zuo. Neural blind deconvolution using deep priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3341–3350, 2020.

[133] Marien Renaud, Jiaming Liu, Valentin de Bortoli, Andrés Almansa, and Ulugbek S Kamilov. Plug-and-play posterior sampling under mismatched measurement and prior models. *arXiv preprint arXiv:2310.03546*, 2023.

[134] Jose A Rodriguez, Rui Xu, C-C Chen, Yunfei Zou, and Jianwei Miao. Oversampling smoothness: an effective algorithm for phase retrieval of noisy diffraction intensities. *Journal of applied crystallography*, 46(2):312–318, 2013.

[135] 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.

[136] Litu Rout, Yujia Chen, Abhishek Kumar, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Beyond first-order tweedie: Solving inverse problems using latent diffusion. *arXiv preprint arXiv:2312.00852*, 2023.

[137] Litu Rout, Advait Parulekar, Constantine Caramanis, and Sanjay Shakkottai. A theoretical justification for image inpainting using denoising diffusion probabilistic models. *arXiv preprint arXiv:2302.01217*, 2023.

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

[139] Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alexandros G Dimakis, and Sanjay Shakkottai. Solving linear inverse problems provably via posterior sampling with latent diffusion models. *arXiv preprint arXiv:2307.00619*, 2023.

[140] 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 Conference Proceedings*, pages 1–10, 2022.

[141] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

[142] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.

[143] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

[144] Masao Shimizu, Shin Yoshimura, Masayuki Tanaka, and Masatoshi Okutomi. Super-resolution from image sequence under influence of hot-air optical turbulence. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[145] Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020.

[146] Emil Y Sidky and Xiaochuan Pan. Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. *Physics in Medicine & Biology*, 53(17):4777, 2008.

[147] Slavko Simic. On a global upper bound for jensen's inequality. *Journal of mathematical analysis and applications*, 343(1):414–419, 2008.

[148] 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.

[149] Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving inverse problems with latent diffusion models via hard data consistency. *arXiv preprint arXiv:2307.08123*, 2023.

[150] Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving inverse problems with latent diffusion models via hard data consistency. In *The Twelfth International Conference on Learning Representations*, 2024.

[151] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR*, 2021.

[152] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023.

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

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

[155] Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations*, 2022.

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

[157] Anuroop Sriram, Jure Zbontar, Tullie Murrell, Aaron Defazio, C Lawrence Zitnick, Nafissa Yakubova, Florian Knoll, and Patricia Johnson. End-to-end variational networks for accelerated MRI reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 64–73. Springer, 2020.

[158] Jiaqi Sun, Alireza Entezari, and Baba C Vemuri. Exploiting structural redundancy in q-space for improved eap reconstruction from highly undersampled (k, q)-space in dmri. *Medical image analysis*, 54:122–137, 2019.

[159] Libin Sun, Sunghyun Cho, Jue Wang, and James Hays. Edge-based blur kernel estimation using patch priors. In *IEEE international conference on computational photography (ICCP)*, pages 1–8. IEEE, 2013.

[160] Phong Tran, Anh Tuan Tran, Quynh Phung, and Minh Hoai. Explore image deblurring via encoded blur kernel space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11956–11965, 2021.

[161] Martin Uecker, Peng Lai, Mark J Murphy, Patrick Virtue, Michael Elad, John M Pauly, Shreyas S Vasanawala, and Michael Lustig. ESPIRiT—an eigenvalue approach to autocalibrating parallel MRI: where SENSE meets GRAPPA. *Magnetic resonance in medicine*, 71(3):990–1001, 2014.

[162] 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.

[163] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in neural information processing systems*, 34:11287–11302, 2021.

[164] Vladimir Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991.

[165] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[166] Bram Wallace, Akash Gokul, and Nikhil Naik. EDICT: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22532–22541, 2023.

[167] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *arXiv preprint arXiv:2305.07015*, 2023.

[168] Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9773–9783, 2023.

[169] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018.

[170] Yinhuai Wang, Jiwen Yu, Runyi Yu, and Jian Zhang. Unlimited-size diffusion restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1160–1167, 2023.

[171] Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *The Eleventh International Conference on Learning Representations*, 2023.

[172] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023.

[173] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36, 2024.

[174] Tianxing Wu, Chenyang Si, Yuming Jiang, Ziqi Huang, and Ziwei Liu. Freeinit: Bridging initialization gap in video diffusion models. *arXiv preprint arXiv:2312.07537*, 2023.

[175] Jaejun Yoo, Kyong Hwan Jin, Harshit Gupta, Jerome Yerly, Matthias Stuber, and Michael Unser. Time-Dependent Deep Image Prior for Dynamic MRI. *IEEE Transactions on Medical Imaging*, 2021.

[176] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14821–14831, 2021.

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

[178] Hanming Zhang, Liang Li, Kai Qiao, Linyuan Wang, Bin Yan, Lei Li, and Guoen Hu. Image prediction for limited-angle tomography via deep learning with convolutional neural network. *arXiv preprint arXiv:1607.08707*, 2016.

[179] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017.

[180] 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.

[181] Ellen D Zhong, Tristan Bepler, Bonnie Berger, and Joseph H Davis. Cryodrgn: reconstruction of heterogeneous cryo-em structures using neural networks. *Nature methods*, 18(2):176–185, 2021.

[182] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

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

Chapter A. Appendix

A.1 Proofs

Theorem 1 (Tweedie's theorem)**.** *Given a Gaussian perturbation kernel $p(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; s_t\boldsymbol{x}_0, \sigma_t^2\boldsymbol{I})$, the posterior mean is given by*

$$\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t] = \frac{1}{s_t}(\boldsymbol{x}_t + \sigma_t^2\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t)) \tag{2.19}$$

Proof.

$$\begin{aligned}
\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t) &= \frac{\nabla_{\boldsymbol{x}_t}p(\boldsymbol{x}_t)}{p(\boldsymbol{x}_t)} && \text{(A.1)}\\
&= \frac{1}{p(\boldsymbol{x}_t)}\nabla_{\boldsymbol{x}_t}\int p(\boldsymbol{x}_t|\boldsymbol{x}_0)p(\boldsymbol{x}_0)\,d\boldsymbol{x}_0 && \text{(A.2)}\\
&= \frac{1}{p(\boldsymbol{x}_t)}\int \nabla_{\boldsymbol{x}_t}p(\boldsymbol{x}_t|\boldsymbol{x}_0)p(\boldsymbol{x}_0)\,d\boldsymbol{x}_0 && \text{(A.3)}\\
&= \frac{1}{p(\boldsymbol{x}_t)}\int p(\boldsymbol{x}_t|\boldsymbol{x}_0)\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t|\boldsymbol{x}_0)p(\boldsymbol{x}_0)\,d\boldsymbol{x}_0 && \text{(A.4)}\\
&= \int p(\boldsymbol{x}_0|\boldsymbol{x}_t)\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t|\boldsymbol{x}_0)\,d\boldsymbol{x}_0 && \text{(A.5)}\\
&= \int p(\boldsymbol{x}_0|\boldsymbol{x}_t)\frac{s_t\boldsymbol{x}_0 - \boldsymbol{x}_t}{\sigma_t^2}\,d\boldsymbol{x}_0 && \text{(A.6)}\\
&= \frac{s_t\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t] - \boldsymbol{x}_t}{\sigma_t^2}. && \text{(A.7)}
\end{aligned}$$

Rearranging the terms, we achieve the conclusion. □

Theorem 5 (Tweedie's theorem for general exponential family)**.** *Let $p(\boldsymbol{y}|\boldsymbol{\eta})$ belong to the exponential family distribution*

$$p(\boldsymbol{y}|\boldsymbol{\eta}) = p_0(\boldsymbol{y})\exp(\boldsymbol{\eta}^\top T(\boldsymbol{y}) - \varphi(\boldsymbol{\eta})), \tag{A.8}$$

where $\boldsymbol{\eta}$ is the canonical vector of the family, $T(\boldsymbol{y})$ is some function of $\boldsymbol{y}$, and $\varphi(\boldsymbol{\eta})$ is the cumulant generation function which normalizes the density, and $p_0(\boldsymbol{y})$ is the density up to the scale factor when $\boldsymbol{\eta} = \boldsymbol{0}$. Then, the posterior mean $\hat{\boldsymbol{\eta}} := \mathbb{E}[\boldsymbol{\eta}|\boldsymbol{y}]$ should satisfy

$$(\nabla_{\boldsymbol{y}}T(\boldsymbol{y}))^\top\hat{\boldsymbol{\eta}} = \nabla_{\boldsymbol{y}}\log p(\boldsymbol{y}) - \nabla_{\boldsymbol{y}}\log p_0(\boldsymbol{y}) \tag{A.9}$$

Proof. Marginal distribution $p(\boldsymbol{y})$ could be expressed as

$$\begin{aligned}
p(\boldsymbol{y}) &= \int p(\boldsymbol{y}|\boldsymbol{\eta})p(\boldsymbol{\eta})d\boldsymbol{\eta} && \text{(A.10)}\\
&= \int p_0(\boldsymbol{y})\exp\left(\boldsymbol{\eta}^\top T(\boldsymbol{y}) - \varphi(\boldsymbol{\eta})\right)p(\boldsymbol{\eta})d\boldsymbol{\eta}. && \text{(A.11)}
\end{aligned}$$

Then, the derivative of the marginal distribution $p(\boldsymbol{y})$ with respect to $\boldsymbol{y}$ becomes

$$\begin{aligned}\nabla_{\boldsymbol{y}} p(\boldsymbol{y}) &= \nabla_y p_0(\boldsymbol{y}) \int \exp\big(\boldsymbol{\eta}^\top T(\boldsymbol{y}) - \varphi(\boldsymbol{\eta})\big) p(\boldsymbol{\eta}) d\boldsymbol{\eta} + \int (\nabla_{\boldsymbol{y}} T(\boldsymbol{y}))^\top \boldsymbol{\eta} p_0(\boldsymbol{y}) \exp\big(\boldsymbol{\eta}^\top T(\boldsymbol{y}) - \varphi(\boldsymbol{\eta})\big) p(\boldsymbol{\eta}) d\boldsymbol{\eta} \\ &= \frac{\nabla_y p_0(\boldsymbol{y})}{p_0(\boldsymbol{y})} \int p(\boldsymbol{y}|\boldsymbol{\eta}) p(\boldsymbol{\eta}) d\boldsymbol{\eta} + (\nabla_{\boldsymbol{y}} T(\boldsymbol{y}))^\top \int \boldsymbol{\eta} p(\boldsymbol{y}|\boldsymbol{\eta}) p(\boldsymbol{\eta}) d\boldsymbol{\eta} \\ &= \frac{\nabla_y p_0(\boldsymbol{y})}{p_0(\boldsymbol{y})} p(\boldsymbol{y}) + (\nabla_{\boldsymbol{y}} T(\boldsymbol{y}))^\top \int \boldsymbol{\eta} p(\boldsymbol{y}, \boldsymbol{\eta}) d\boldsymbol{\eta}\end{aligned}$$

Therefore,

$$\frac{\nabla_y p(\boldsymbol{y})}{p(\boldsymbol{y})} = \frac{\nabla_y p_0(\boldsymbol{y})}{p_0(\boldsymbol{y})} + (\nabla_{\boldsymbol{y}} T(\boldsymbol{y}))^\top \int \boldsymbol{\eta} p(\boldsymbol{\eta}|\boldsymbol{y}) d\boldsymbol{\eta} \tag{A.12}$$

which is equivalent to

$$(\nabla_{\boldsymbol{y}} T(\boldsymbol{y}))^\top \mathbb{E}[\boldsymbol{\eta}|\boldsymbol{y}] = \nabla_{\boldsymbol{y}} \log p(\boldsymbol{y}) - \nabla_{\boldsymbol{y}} \log p_0(\boldsymbol{y}) \tag{A.13}$$

This concludes the proof. □

Proposition 4 (Jensen gap upper bound [53])**.** *Define the absolute cenetered moment as* $m_p := \sqrt[p]{\mathbb{E}[|X - \mu|^p]}$*, and the mean as* $\mu = \mathbb{E}[X]$*. Assume that for* $\alpha > 0$*, there exists a positive number* K *such that for any* $x \in \mathbb{R}, |f(x) - f(\mu)| \leq K|x - \mu|^\alpha$*. Then,*

$$|\mathbb{E}[f(X) - f(\mathbb{E}[X])]| \leq \int |f(X) - f(\mu)| dp(X) \tag{A.14}$$

$$\leq K \int |x - \mu|^\alpha dp(X) \leq M m_\alpha^\alpha. \tag{A.15}$$

Lemma 1. *Let* $\phi(\cdot)$ *be a univariate Gaussian density function with mean* μ *and variance* σ^2*. There exists a constant* L *such that* $\forall x, y \in \mathbb{R}$*,*

$$|\phi(x) - \phi(y)| \leq L|x - y|, \tag{A.16}$$

where $L = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{1}{2\sigma^2})$.

Proof. As ϕ' is continuous and bounded, we use the mean value theorem to get

$$\forall (x, y) \in \mathbb{R}^2, |\phi(x) - \phi(y)| \leq \|\phi'\|_\infty |x - y|. \tag{A.17}$$

Since L is the minimal value for Eq. (A.16), we have that $L \leq \|\phi'\|_\infty$. Taking the limit $y \to x$ gives $|\phi'(x)| \leq L$, and thus $\|\phi'\|_\infty \leq L$. Hence

$$L = \|\phi'\|_\infty = \| - \frac{x - \mu}{\sigma^2} \phi(x) \|_\infty. \tag{A.18}$$

Since the derivative of ϕ' is given as

$$\phi''(x) = \sigma^{-2}(1 - \sigma^{-2}(x - \mu)^2)\phi(x), \tag{A.19}$$

and the maximum is attained when $x = 1 \pm \sigma^2\mu$, we have

$$L = \|\phi'\|_\infty = \frac{e^{-1/2\sigma^2}}{\sqrt{2\pi\sigma^2}} \tag{A.20}$$

□

Lemma 2. *Let $\phi(\cdot)$ be an isotropic multivariate Gaussian density function with mean $\boldsymbol{\mu}$ and variance $\sigma^2\boldsymbol{I}$. There exists a constant L such that $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$,*

$$\|\phi(\boldsymbol{x}) - \phi(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|, \tag{A.21}$$

where $L = \frac{d}{\sqrt{2\pi\sigma^2}} e^{-1/2\sigma^2}$.

Proof.

$$\|\phi(\boldsymbol{x}) - \phi(\boldsymbol{y})\| \leq \max_{\boldsymbol{z}} \|\nabla_{\boldsymbol{z}} \boldsymbol{\phi}(\boldsymbol{z})\| \cdot \|\boldsymbol{x} - \boldsymbol{y}\| \tag{A.22}$$

$$= \underbrace{\frac{d}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\right)}_{L} \cdot \|\boldsymbol{x} - \boldsymbol{y}\| \tag{A.23}$$

where the second inequality comes from that each element of $\nabla_{\boldsymbol{z}} \boldsymbol{\phi}(\boldsymbol{z})$ is bounded by $\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\right)$. □

Theorem 2. *For the given measurement model Eq. (2.1) with $\boldsymbol{n} \sim \mathcal{N}(0, \sigma_y^2\boldsymbol{I})$, we have*

$$p(\boldsymbol{y}|\boldsymbol{x}_t) \simeq p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0), \tag{3.8}$$

where the approximation error can be quantified with the Jensen gap, which is upper bounded by

$$\mathcal{J} \leq \frac{d}{\sqrt{2\pi\sigma_y^2}} e^{-1/2\sigma_y^2} \|\nabla_{\boldsymbol{x}} \mathcal{A}(\boldsymbol{x})\| m_1, \tag{3.9}$$

where $\|\nabla_{\boldsymbol{x}} \mathcal{A}(\boldsymbol{x})\| := \max_{\boldsymbol{x}} \|\nabla_{\boldsymbol{x}} \mathcal{A}(\boldsymbol{x})\|$ and $m_1 := \int \|\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0\| p(\boldsymbol{x}_0|\boldsymbol{x}_t)\, d\boldsymbol{x}_0$.

Proof.

$$p(\boldsymbol{y}|\boldsymbol{x}_t) = \int p(\boldsymbol{y}|\boldsymbol{x}_0) p(\boldsymbol{x}_0|\boldsymbol{x}_t) d\boldsymbol{x}_0 \tag{A.24}$$

$$= \mathbb{E}_{\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0|\boldsymbol{x}_t)}[f(\boldsymbol{x}_0)] \tag{A.25}$$

Here, $f(\cdot) := h(\mathcal{A}(\cdot))$ where $\mathcal{A}$ is the forward operator and $h(\boldsymbol{x})$ is the multivariate normal distribution with mean

$\boldsymbol{y}$ and the covariance $\sigma_y^2 \boldsymbol{I}$. Therefore, we have

$$J(f, p(\boldsymbol{x}_0|\boldsymbol{x}_t)) = |\mathbb{E}[f(\boldsymbol{x}_0)] - f(\mathbb{E}[\boldsymbol{x}_0])| = |\mathbb{E}[f(\boldsymbol{x}_0)] - f(\hat{\boldsymbol{x}}_0)| \tag{A.26}$$

$$= |\mathbb{E}[h(\mathcal{A}(\boldsymbol{x}_0))] - h(\mathcal{A}(\hat{\boldsymbol{x}}_0))| \tag{A.27}$$

$$\leq \int |h(\mathcal{A}(\boldsymbol{x}_0)) - h(\mathcal{A}(\hat{\boldsymbol{x}}_0))| dP(\boldsymbol{x}_0|\boldsymbol{x}_t) \tag{A.28}$$

$$\overset{\text{(b)}}{\leq} \frac{d}{\sqrt{2\pi\sigma_y^2}} e^{-1/2\sigma_y^2} \int \|\mathcal{A}(\boldsymbol{x}_0) - \mathcal{A}(\hat{\boldsymbol{x}}_0)\| dP(\boldsymbol{x}_0|\boldsymbol{x}_t) \tag{A.29}$$

$$\overset{\text{(c)}}{\leq} \frac{d}{\sqrt{2\pi\sigma_y^2}} e^{-1/2\sigma_y^2} \|\nabla_{\boldsymbol{x}} \mathcal{A}(\boldsymbol{x})\| \int \|\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0\| dP(\boldsymbol{x}_0|\boldsymbol{x}_t) \tag{A.30}$$

$$\overset{\text{(d)}}{\leq} \frac{d}{\sqrt{2\pi\sigma_y^2}} e^{-1/2\sigma_y^2} \|\nabla_{\boldsymbol{x}} \mathcal{A}(\boldsymbol{x})\| m_1 \tag{A.31}$$

where $dP(\boldsymbol{x}_0|\boldsymbol{x}_t) = p(\boldsymbol{x}_0|\boldsymbol{x}_t)\, d\boldsymbol{x}_0$, (b) is the result of Lemma 2, (c) is from the intermediate value theorem, and (d) is from Proposition 4. □

Theorem 3 (informal)**.** *With similar approximation error as in Theorem 2,*

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t, \boldsymbol{k}_t) \simeq \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t), \hat{\boldsymbol{k}}_0(\boldsymbol{k}_t))$$
$$\nabla_{\boldsymbol{k}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t, \boldsymbol{k}_t) \simeq \nabla_{\boldsymbol{k}_t} \log p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t), \hat{\boldsymbol{k}}_0(\boldsymbol{k}_t)).$$

Proof. We first note that $\boldsymbol{x}_t, \boldsymbol{k}_t \, \forall t \in [0, 1]$ are independent. Further, $\boldsymbol{y}$ and $\boldsymbol{x}_t$ are conditionally independent on $\boldsymbol{x}_0$; $\boldsymbol{y}$ and $\boldsymbol{k}_t$ are conditionally independent on $\boldsymbol{k}_0$. Then, we have the following factorization

$$p(\boldsymbol{y}|\boldsymbol{x}_t, \boldsymbol{k}_t) = \int p(\boldsymbol{y}|\boldsymbol{x}_0, \boldsymbol{k}_0) p(\boldsymbol{x}_0|\boldsymbol{x}_t) p(\boldsymbol{k}_0|\boldsymbol{k}_t)\, d\boldsymbol{x}_0 d\boldsymbol{k}_0 \tag{A.32}$$

$$= \mathbb{E}_{\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0|\boldsymbol{x}_t),\, \boldsymbol{k}_0 \sim p(\boldsymbol{k}_0|\boldsymbol{k}_t)}[f(\boldsymbol{x}_0, \boldsymbol{k}_0)], \tag{A.33}$$

where $f(\boldsymbol{x}_0, \boldsymbol{k}_0) = h(\boldsymbol{k}_0 * \boldsymbol{x}_0)$, with $h(\boldsymbol{\mu})$ denoting the density function of an isotropic multivariate Gaussian density function with mean $\boldsymbol{\mu}$, and variance $\sigma^2 \boldsymbol{I}$. Our proposal is to use the Jensen approximation

$$p(\boldsymbol{y}|\boldsymbol{x}_t, \boldsymbol{k}_t) \simeq p(\boldsymbol{y}|\mathbb{E}[\boldsymbol{x}_0, \boldsymbol{k}_0]) = p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0, \hat{\boldsymbol{k}}_0), \tag{A.34}$$

where the last equality comes from the independency of $\boldsymbol{x}_0$ and $\boldsymbol{k}_0$. Now we derive the closed-form upper bound of the Jensen gap. For simplicity in exposition, let us define $\boldsymbol{K}_0 \boldsymbol{x}_0 \equiv \boldsymbol{k}_0 * \boldsymbol{x}_0, \equiv \boldsymbol{X}_0 \boldsymbol{k}_0$, where $\boldsymbol{K}_0, \boldsymbol{X}_0$ are block Hankel matrices that represent the convolution operation in matrix multiplication. Further, we denote $\|\bar{\boldsymbol{K}_0}\| := \mathbb{E}_{\boldsymbol{k}_0 \sim p(\boldsymbol{k}_0|\boldsymbol{k}_t)}[\|\boldsymbol{K}_0\|]$. Our Jensen gap reads

$$\mathcal{J}(f, p(\boldsymbol{x}_0|\boldsymbol{x}_t) p(\boldsymbol{k}_0|\boldsymbol{k}_t)) = |\mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{k}_0}[f(\boldsymbol{x}_0, \boldsymbol{k}_0)] - f(\mathbb{E}_{\boldsymbol{x}_0}[\boldsymbol{x}_0], \mathbb{E}_{\boldsymbol{k}_0}[\boldsymbol{k}_0])| \tag{A.35}$$

$$\leq \underbrace{|\mathbb{E}_{\boldsymbol{k}_0, \boldsymbol{x}_0}[f(\boldsymbol{x}_0, \boldsymbol{k}_0)] - \mathbb{E}_{\boldsymbol{k}_0}[f(\mathbb{E}_{\boldsymbol{x}}[\boldsymbol{x}_0], \boldsymbol{k}_0)]|}_{①} + \underbrace{|\mathbb{E}_{\boldsymbol{k}_0}[f(\mathbb{E}_{\boldsymbol{x}_0}[\boldsymbol{x}_0], \boldsymbol{k}_0)] - f(\mathbb{E}[\boldsymbol{x}_0], \mathbb{E}[\boldsymbol{k}_0])|}_{②}, \tag{A.36}$$

with

$$
\begin{aligned}
① &= |\mathbb{E}_{\boldsymbol{k}_0}[\mathbb{E}_{\boldsymbol{x}_0}[f(\boldsymbol{x}_0, \boldsymbol{k}_0)] - f(\mathbb{E}_{\boldsymbol{x}_0}[\boldsymbol{x}_0], \boldsymbol{k}_0)]| && \text{(A.37)}\\
&\overset{\text{(a)}}{\leq} \mathbb{E}_{\boldsymbol{k}_0}\left[\int |h(\boldsymbol{k}_0 * \boldsymbol{x}_0) - h(\boldsymbol{k}_0 * \hat{\boldsymbol{x}}_0)| dP(\boldsymbol{x}_0|\boldsymbol{x}_t)\right] && \text{(A.38)}\\
&\overset{\text{(b)}}{\leq} \mathbb{E}_{\boldsymbol{k}_0}\left[\frac{d}{\sqrt{2\pi\sigma^2}} e^{-1/2\sigma^2} \int \|\boldsymbol{K}_0\boldsymbol{x}_0 - \boldsymbol{K}_0\hat{\boldsymbol{x}}_0\| dP(\boldsymbol{x}_0|\boldsymbol{x}_t)\right] && \text{(A.39)}\\
&\leq \mathbb{E}_{\boldsymbol{k}_0}\left[\frac{d}{\sqrt{2\pi\sigma^2}} e^{-1/2\sigma^2} \|\boldsymbol{K}_0\| \int \|\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0\| dP(\boldsymbol{x}_0|\boldsymbol{x}_t)\right] && \text{(A.40)}\\
&\overset{\text{(c)}}{\leq} \mathbb{E}_{\boldsymbol{k}_0}\left[\frac{d}{\sqrt{2\pi\sigma^2}} e^{-1/2\sigma^2} \|\boldsymbol{K}_0\| m_{1,\boldsymbol{x}_0}\right] && \text{(A.41)}\\
&\overset{\text{(d)}}{\leq} \frac{d}{\sqrt{2\pi\sigma^2}} e^{-1/2\sigma^2} \|\bar{\boldsymbol{K}}_0\| m_{1,\boldsymbol{x}_0}, && \text{(A.42)}
\end{aligned}
$$

where (a) is from Proposition. 4, (b) is from Lemma. 2, and (c-d) are from the definitions. Moreover,

$$
\begin{aligned}
② &\leq \int |h(\hat{\boldsymbol{k}}_0 * \hat{\boldsymbol{x}}_0) - h(\boldsymbol{k}_0 * \hat{\boldsymbol{x}}_0)| dP(\boldsymbol{k}_0|\boldsymbol{k}_t) && \text{(A.43)}\\
&\leq \frac{d}{\sqrt{2\pi\sigma^2}} e^{-1/2\sigma^2} \int \|\hat{\boldsymbol{X}}_0\boldsymbol{k}_0 - \hat{\boldsymbol{X}}_0\hat{\boldsymbol{k}}_0\| dP(\boldsymbol{k}_0|\boldsymbol{k}_t) && \text{(A.44)}\\
&\leq \frac{d}{\sqrt{2\pi\sigma^2}} e^{-1/2\sigma^2} \|\hat{\boldsymbol{X}}_0\| m_{1,\boldsymbol{k}_0}. && \text{(A.45)}
\end{aligned}
$$

Hence

$$
\mathcal{J}(f, p(\boldsymbol{x}_0|\boldsymbol{x}_t)p(\boldsymbol{k}_0|\boldsymbol{k}_t)) \leq \frac{d}{\sqrt{2\pi\sigma^2}} e^{-1/2\sigma^2} \left(\|\bar{\boldsymbol{K}}_0\| m_{1,\boldsymbol{x}_0} + \|\hat{\boldsymbol{X}}_0\| m_{1,\boldsymbol{k}_0}\right). \tag{A.46}
$$

where

$$
\begin{aligned}
m_{1,\boldsymbol{x}_0} &:= \int \|\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0\| dP(\boldsymbol{x}_0|\boldsymbol{x}_t) && \text{(A.47)}\\
m_{1,\boldsymbol{k}_0} &:= \int \|\boldsymbol{k}_0 - \hat{\boldsymbol{k}}_0\| dP(\boldsymbol{k}_0|\boldsymbol{k}_t) && \text{(A.48)}
\end{aligned}
$$

We have derived that the approximation Eq. (A.34) has the Jensen gap upper bounded by Eq. (A.46). Finally, taking the derivative of the log to Eq. (A.34), we have that

$$
\begin{aligned}
\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t, \boldsymbol{k}_t) &\simeq \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t), \hat{\boldsymbol{k}}_0(\boldsymbol{k}_t))\\
\nabla_{\boldsymbol{k}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t, \boldsymbol{k}_t) &\simeq \nabla_{\boldsymbol{k}_t} \log p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t), \hat{\boldsymbol{k}}_0(\boldsymbol{k}_t)).
\end{aligned}
$$

Note that the approximation error from the Jensen gap approaches to zero as the noise level σ increase sufficiently. □

Proposition 1 (Concentration of noisy data)**.** *Consider the distribution of noisy data* $p_i(\boldsymbol{x}_i) = \int p(\boldsymbol{x}_i|\boldsymbol{x}) p_0(\boldsymbol{x}) d\boldsymbol{x}, p(\boldsymbol{x}_i|\boldsymbol{x}) \sim \mathcal{N}(a_i\boldsymbol{x}, b_i^2\boldsymbol{I})$*. Then* $p_i(\boldsymbol{x}_i)$ *is concentrated on* $(n-1)$*-dim manifold* $\mathcal{M}_i := \{\boldsymbol{y} \in \mathbb{R}^n : d(\boldsymbol{y}, a_i\mathcal{M}) = r_i := b_i\sqrt{n-l}\}$*. Rigorously,* $p_i(B_{\epsilon r_i}(\mathcal{M}_i)) > 1 - \delta$*, for some small* $\epsilon, \delta > 0$*.*

Proof. Suppose that the data manifold is an l-dimensional linear subspace. By rotation and translation, we safely assume that $\mathcal{M} = \{\boldsymbol{x} \in \mathbb{R}^n : x_{l+1} = x_{l+2} = \cdots = x_n = 0\}$. Then, we can simply write $d(\boldsymbol{x}, \mathcal{M}) = \sqrt{x_{l+1}^2 + \cdots + x_n^2}$, and $\mathcal{M}_i = \{\boldsymbol{x} \in \mathbb{R}^n : x_{l+1}^2 + \cdots + x_n^2 = r_i^2\}$. For a given point $\boldsymbol{x}' = (x_1', x_2', \dots) \in \mathcal{M}$, we consider $p(\boldsymbol{x}|\boldsymbol{x}') \sim \mathcal{N}(a_i\boldsymbol{x}', b_i^2 I)$ and obtain a concentration inequality independent to the choice of $\boldsymbol{x}'$. We need

the standard Laurent-Massart bound for a chi-square variable [98]. When X is a chi-square distribution with k degrees of freedom,

$$P[X - k \geq 2\sqrt{kt} + 2t] \leq e^{-t},$$
$$P[X - k \leq -2\sqrt{kt}] \leq e^{-t}.$$

As $\frac{x_{l+1}^2}{b_i^2} + \cdots + \frac{x_n^2}{b_i^2}$ is a chi-square distribution with $n - l$ degrees of freedom, by substituting $t = (n - l)\varepsilon'$ in the above bound,

$$P\left[-2(n-l)\sqrt{\varepsilon'} \leq \frac{x_{l+1}^2}{b_i^2} + \cdots + \frac{x_n^2}{b_i^2} - (n-l) \leq 2(n-l)(\sqrt{\varepsilon'} + \varepsilon')\right]$$
$$= P\left[\sqrt{x_{l+1}^2 + \cdots + x_n^2} \in (\, r_i\sqrt{1 - 2\sqrt{\varepsilon'}},\, r_i\sqrt{1 + 2\sqrt{\varepsilon'} + 2\varepsilon'}\,)\right] \geq 1 - 2e^{-(n-l)\varepsilon'}.$$

Note that the above inequality does not depend on $x_1, \ldots x_l$, thus the choice of $\boldsymbol{x}' \in \mathcal{M}$. As a result, by setting $\varepsilon = \min\{1 - \sqrt{1 - 2\sqrt{\varepsilon'}}, \sqrt{1 + 2\sqrt{\varepsilon'} + 2\varepsilon'} - 1\}$ and $\delta = 2e^{-(n-l)\varepsilon'}$,

$$p(\boldsymbol{x} \in B_{\varepsilon r_i}(M_i)|\boldsymbol{x}') > 1 - \delta,$$

thus

$$p_i(\boldsymbol{x} \in B_{\varepsilon r_i}(M_i)) = \int p(\boldsymbol{x} \in B_{\varepsilon r_i}(M_i)|\boldsymbol{x}')p(\boldsymbol{x}')d\boldsymbol{x}' > 1 - \delta.$$

□

Proposition 2 (score function)**.** *Suppose $\boldsymbol{s}_\theta$ is the minimizer of the denoising score matching loss in Eq. (2.17). Let Q_i be the function that maps $\boldsymbol{x}_i$ to $\hat{\boldsymbol{x}}_0$ for each i,*

$$Q_i : \mathbb{R}^d \to \mathbb{R}^d, \boldsymbol{x}_i \mapsto \hat{\boldsymbol{x}}_0 := \frac{1}{a_i}(\boldsymbol{x}_i + b_i^2 \boldsymbol{s}_\theta(\boldsymbol{x}_i, i)).$$

Then, $Q_i(\boldsymbol{x}_i) \in \mathcal{M}$ and $\boldsymbol{J}_{Q_i}^2 = \boldsymbol{J}_{Q_i} = \boldsymbol{J}_{Q_i}^T : \mathbb{R}^d \to T_{Q_i(\boldsymbol{x}_i)}\mathcal{M}$. Intuitively, Q_i is locally an orthogonal projection onto $\mathcal{M}$.

Proof. To minimize Eq. (2.17), or equivalently,

$$\int ||\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{x}_0)||_2^2 p(\boldsymbol{x}_t|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}d\boldsymbol{x}_t dt,$$

By differentiating the objective with respect to $s_\theta(\boldsymbol{x}_t, t)$, we have

$$\int \left(\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \frac{a_t\boldsymbol{x} - \boldsymbol{x}_t}{b_t^2}\right) p(\boldsymbol{x}_t|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} = 0$$
$$\int \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)p(\boldsymbol{x}_t)p(\boldsymbol{x}|\boldsymbol{x}_t)d\boldsymbol{x} = \int \frac{a_t\boldsymbol{x} - \boldsymbol{x}_t}{b_t^2} p(\boldsymbol{x}_t)p(\boldsymbol{x}|\boldsymbol{x}_t)d\boldsymbol{x}$$
$$\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) \int p(\boldsymbol{x}|\boldsymbol{x}_t)d\boldsymbol{x} = \int \frac{a_t\boldsymbol{x}}{b_t^2} p(\boldsymbol{x}|\boldsymbol{x}_t)d\boldsymbol{x} - \frac{\boldsymbol{x}_t}{b_t^2} \int p(\boldsymbol{x}|\boldsymbol{x}_t)d\boldsymbol{x}$$
$$\therefore \boldsymbol{s}_\theta(\boldsymbol{x}_t, t) = \frac{1}{b_t^2}(-\boldsymbol{x}_t + a_t \int \boldsymbol{x}p(\boldsymbol{x}|\boldsymbol{x}_t)d\boldsymbol{x}) \forall \boldsymbol{x}_t, t,$$

where we used $p(\boldsymbol{x}_t|\boldsymbol{x})p(\boldsymbol{x}) = p(\boldsymbol{x}, \boldsymbol{x}_t) = p(\boldsymbol{x}_t)p(\boldsymbol{x}|\boldsymbol{x}_t)$, $p(\boldsymbol{x}_t) > 0$, and $\int p(\boldsymbol{x}|\boldsymbol{x}_t)d\boldsymbol{x} = 1$ in each line. Here, $Q_i(\boldsymbol{x}_i) = \int \boldsymbol{x}p(\boldsymbol{x}|\boldsymbol{x}_i)d\boldsymbol{x}$ is the weighted average vector of points on the data manifold as $p(\boldsymbol{x}|\boldsymbol{x}_i)$ is supported on

the data manifold. Combining it with the assumption that the manifold is linear, $Q_i(\boldsymbol{x}_i) \in \mathcal{M}$.

Considering the symmetry of $p(\boldsymbol{x}|\boldsymbol{x}_i)$ about $\boldsymbol{x}_i$, $p(\boldsymbol{x}|\boldsymbol{x}_i)$ is a radial function on $\mathcal{M}$, centering around the nearest point to $\boldsymbol{x}_i$ on $\mathcal{M}$. Hence, $Q_i(\boldsymbol{x}_i)$ shall be the nearest point to $\boldsymbol{x}_i$ of all points on $\mathcal{M}$. Therefore, J_{Q_i} is the orthogonal projection onto $T_{Q_i(\boldsymbol{x}_i)}\mathcal{M}$. Stating more rigorously, let $\boldsymbol{u} = \boldsymbol{u}_t + \boldsymbol{u}_n \in \mathbb{R}^n$ for $\boldsymbol{u}_t \in T_{Q_i(\boldsymbol{x}_i)}\mathcal{M}, \boldsymbol{u}_n \perp T_{Q_i(\boldsymbol{x}_i)}\mathcal{M}$. Then, for a scalar s, $Q_i(\boldsymbol{x}_i + s\boldsymbol{u}) = Q_i(\boldsymbol{x}_i) + s\boldsymbol{u}_t$, as only tangent component to the manifold change the nearest point. By differentiating with respect to s, we obtain $\boldsymbol{J}_{Q_i}\boldsymbol{u} = \boldsymbol{u}_t$, thus $\boldsymbol{J}_{Q_i}^2 = \boldsymbol{J}_{Q_i}$. For another vector $\boldsymbol{v} = \boldsymbol{v}_t + \boldsymbol{v}_n$ with $\boldsymbol{v}_t \in T_{Q_i(\boldsymbol{x}_i)}\mathcal{M}, \boldsymbol{v}_n \perp T_{Q_i(\boldsymbol{x}_i)}\mathcal{M}$,

$$\begin{aligned}\boldsymbol{v}^T\boldsymbol{J}_{Q_i}\boldsymbol{u} &= (\boldsymbol{v}_t + \boldsymbol{v}_n)^T\boldsymbol{u}_t \\ &= \boldsymbol{v}_t^T\boldsymbol{u}_t \\ &= (\boldsymbol{u}_t + \boldsymbol{u}_n)^T\boldsymbol{v}_t \\ &= \boldsymbol{u}^T\boldsymbol{J}_{Q_i}\boldsymbol{v},\end{aligned}$$

where we applied $\boldsymbol{v}_n^T\boldsymbol{u}_t = 0 = \boldsymbol{u}_n^T\boldsymbol{v}_t$. Therefore, $\boldsymbol{J}_{Q_i}$ is symmetric, i.e. $\boldsymbol{J}_{Q_i}^T = \boldsymbol{J}_{Q_i}$, which concludes this proof. □

Theorem 4 (Manifold constrained gradient)**.** *A correction by the manifold constrained gradient does not leave the data manifold. Formally,*

$$\frac{\partial}{\partial \boldsymbol{x}_i}\|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0\|_2^2 = -2\boldsymbol{J}_{Q_i}^T\boldsymbol{A}^T(\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0) \in T_{\hat{\boldsymbol{x}}_0}\mathcal{M},$$

the gradient is the projection of the data fidelity term onto $T_{\hat{\boldsymbol{x}}_0}\mathcal{M}$,

Proof.

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{x}_i}\|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0\|_2^2 &= -2\boldsymbol{J}_{\boldsymbol{A}Q_i}^T(\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0) \\ &= -2\boldsymbol{J}_{Q_i}^T\boldsymbol{A}^T(\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0) \\ &= \boldsymbol{J}_{Q_i}d \in T_{Q_i(\boldsymbol{x}_i)}\mathcal{M}\end{aligned}$$

where $d = -2\boldsymbol{A}^T(\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_0)$. The first and second equality is given by the chain rule and the last line is by Proposition 2. □

A.2 Mathematical Background

A.2.1 ADMM

In this section, we derive the ADMM-TV optimization framework for completeness. We are interested in solving the problem of TV-regularized WLS of the following form:

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{D}_z\boldsymbol{x}\|_1, \tag{A.49}$$

where $\boldsymbol{D}_z$ takes the finite difference across the $z-$dimension. In order to solve the problem in an alternating fashion, we split the variables

$$\min_{\boldsymbol{x},\boldsymbol{z}} \quad \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2+\lambda\|\boldsymbol{z}\|_1 \tag{A.50}$$

$$\text{s.t.} \quad \boldsymbol{z}=\boldsymbol{D}_z\boldsymbol{x}. \tag{A.51}$$

The scaled formulation of ADMM [11] is then given by

$$\boldsymbol{x}^+=\arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2+\frac{\rho}{2}\|\boldsymbol{D}_z\boldsymbol{x}-\boldsymbol{z}+\boldsymbol{w}\|_2^2 \tag{A.52}$$

$$\boldsymbol{z}^+=\arg\min_{\boldsymbol{z}} \lambda\|\boldsymbol{z}\|_1+\frac{\rho}{2}\|\boldsymbol{D}_z\boldsymbol{x}^+-\boldsymbol{z}+\boldsymbol{w}\|_2^2 \tag{A.53}$$

$$\boldsymbol{w}^+=\boldsymbol{w}+\boldsymbol{D}_z\boldsymbol{x}^+-\boldsymbol{z}^+. \tag{A.54}$$

Eq. (A.52) is convex and smooth and thus has a closed-form solution

$$\boldsymbol{x}^+=(\boldsymbol{A}^T\boldsymbol{A}+\rho\boldsymbol{D}_z^T\boldsymbol{D}_z)^{-1}(\boldsymbol{A}^T\boldsymbol{y}+\rho\boldsymbol{D}_z^T(\boldsymbol{z}-\boldsymbol{w})), \tag{A.55}$$

where one can perform CG rather than computing the matrix inverse directly. In order to solve, Eq. (A.53), we define the proximal operator [123] as

$$\text{prox}_{f,\eta}(\boldsymbol{z}) \triangleq \arg\min_{\boldsymbol{x}} f(\boldsymbol{x})+\frac{1}{2\eta}\|\boldsymbol{x}-\boldsymbol{z}\|_2^2. \tag{A.56}$$

By inspecting Eq. (A.53), we know that it is in the form of proximal mapping

$$\boldsymbol{z}^+=\text{prox}_{\|\cdot\|_1,\lambda/\rho}(\boldsymbol{D}_z\boldsymbol{x}+\boldsymbol{w}) \tag{A.57}$$

$$=\mathcal{S}_{\lambda/\rho}(\boldsymbol{D}_z\boldsymbol{x}+\boldsymbol{w}), \tag{A.58}$$

where we have leveraged the fact that the proximal mapping of the ℓ_1 norm is given as the soft thresholding operator $\mathcal{S}$. In summary, we have

$$\begin{aligned}
\boldsymbol{x}^+&=(\boldsymbol{A}^T\boldsymbol{A}+\rho\boldsymbol{D}_z^T\boldsymbol{D}_z)^{-1}(\boldsymbol{A}^T\boldsymbol{y}+\rho\boldsymbol{D}_z^T(\boldsymbol{z}-\boldsymbol{w}))\\
\boldsymbol{z}^+&=\mathcal{S}_{\lambda/\rho}(\boldsymbol{D}_z\boldsymbol{x}^+ +\boldsymbol{w})\\
\boldsymbol{w}^+&=\boldsymbol{w}+\boldsymbol{D}_z\boldsymbol{x}^+-\boldsymbol{z}^+.
\end{aligned}$$

A.2.2 Krylov subspace methods

Consider the linear system $\boldsymbol{y}=\boldsymbol{A}\boldsymbol{x}$. In classical projection based methods such as Krylov subspace methods [102], for given two subspace $\mathcal{K}$ and $\mathcal{L}$, we define an approximate problem to find $\boldsymbol{x}\in\mathcal{K}$ such that

$$\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\perp\mathcal{L} \tag{A.59}$$

This is a basic projection step, and the sequence of such steps is applied. For example, with non-zero estimate $\hat{\boldsymbol{x}}$, the associated problem is to find $\boldsymbol{x}\in\hat{\boldsymbol{x}}+\mathcal{K}$ such that $\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\perp\mathcal{L}$, which is equivalent to finding $\boldsymbol{\delta}\in\mathcal{K}$ such that

$$\boldsymbol{b}-\boldsymbol{A}\boldsymbol{\delta}\perp\mathcal{L}, \quad \boldsymbol{\delta}:=\boldsymbol{x}-\hat{\boldsymbol{x}}, \quad \boldsymbol{b}:=\boldsymbol{y}-\boldsymbol{A}\hat{\boldsymbol{x}}. \tag{A.60}$$

In terms of the choice of the two subspaces, the CG method chooses the two subspaces $\mathcal{K}$ and $\mathcal{L}$ as the same Krylov subspace:

$$\mathcal{K} = \mathcal{L} = \mathcal{K}_l := \text{Span}(\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \cdots, \boldsymbol{A}^{l-1}\boldsymbol{b}). \tag{A.61}$$

Then, CG attempts to find the solution to the following optimization problem:

$$\min_{\boldsymbol{x} \in \hat{\boldsymbol{x}} + \mathcal{K}_l} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 \tag{A.62}$$

Krylov subspace methods can be also extended to nonlinear problems via zero-finding. Specifically, the optimization problem $\min_{\boldsymbol{x}} \ell(\boldsymbol{x})$ can be equivalently converted to a zero-finding problem of its gradient, i.e. $\nabla_{\boldsymbol{x}} \ell(\boldsymbol{x}) = \boldsymbol{0}$. If we consider a non-linear forward imaging operator $\mathcal{A}(\cdot)$, we can define $\ell(\boldsymbol{x}) = \|\boldsymbol{y} - \mathcal{A}(\boldsymbol{x})\|^2/2$. Then, one can use, for example, Newton-Krylov method [91] to linearize the problem near the current solution and apply standard Krylov methods to solve the current problem. Now, given the optimization problem, we can see the fundamental differences between the gradient-based approaches and Krylov methods. Specifically, gradient methods are based on the iteration:

$$\boldsymbol{x}^{(i+1)} = \boldsymbol{x}^{(i)} - \gamma \nabla_{\boldsymbol{x}} \ell(\boldsymbol{x}^{(i)}) \tag{A.63}$$

which stops updating when $\nabla_{\boldsymbol{x}} \ell(\boldsymbol{x}^{(i)}) \simeq \boldsymbol{0}$. On the other hand, Krylov subspace methods try to find $\boldsymbol{x} \in \mathcal{K}_l$ by increasing l to achieve a better approximation of $\nabla_{\boldsymbol{x}} \ell(\boldsymbol{x}) = \boldsymbol{0}$. This difference allows us to devise a computationally efficient algorithm when combined with diffusion models.

A.3 Application of CFG++ to higher order solvers

In this section, we discuss ways in which we can apply CFG++ to a more diverse set of ODE/SDE solvers. We consider solving the variance exploding (VE) PF-ODE, as re-parametrization VP diffusion models can easily recover the VE formulation, as often implemented in widely used frameworks such as https://github.com/crowsonkb/k-diffusion. Following the notation in [111], we consider a sequence of timesteps $\{t_i\}_{i=0}^M$, where $t_0 = T$ denotes the initial starting point of the reverse sampling (i.e. Gaussian noise).

Euler [79] The construction is the same as in DDIM. We include it here for completeness. The update step reads

$$\boldsymbol{x}_{t_{i+1}} = \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t_i}) + \frac{\boldsymbol{x}_{t_i} - \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t_i})}{\sigma_{t_i}} \cdot \sigma_{t_{i+1}}, \qquad \text{(CFG)}$$

$$\boldsymbol{x}_{t_{i+1}} = \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\lambda}(\boldsymbol{x}_{t_i}) + \frac{\boldsymbol{x}_{t_i} - \hat{\boldsymbol{x}}_{\varnothing}(\boldsymbol{x}_{t_i})}{\sigma_{t_i}} \cdot \sigma_{t_{i+1}}, \qquad \text{(CFG++)}$$

Euler Ancestral Euler Ancestral sampler follows Euler sampler, but introduces stochasticity by taking larger steps and then adding a slight amount of noise. Adjusting with CFG++ is straightforward.

$$\boldsymbol{x}_{t_{i+1}} = \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t_i}) + \frac{\boldsymbol{x}_{t_i} - \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t_i})}{\sigma_{t_i}} \cdot (\sigma_{t_{d_i}} - \sigma_{t_i}) + \sigma_{t_i}\boldsymbol{\epsilon}, \qquad \text{(CFG)}$$

$$\boldsymbol{x}_{t_{i+1}} = \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\lambda}(\boldsymbol{x}_{t_i}) + \frac{\boldsymbol{x}_{t_i} - \hat{\boldsymbol{x}}_{\varnothing}(\boldsymbol{x}_{t_i})}{\sigma_{t_i}} \cdot (\sigma_{t_{d_i}} - \sigma_{t_i}) + \sigma_{t_i}\boldsymbol{\epsilon}, \qquad \text{(CFG++)}$$

where $t_i > t_{d_i} > t_{i+1}$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$.

DPM-solver++ 2M [111] Define $\sigma_t := e^{-t}$, $h_i := t_i - t_{i-1}$, and $r_i := h_{i-1}/h_i$. After initializing $\boldsymbol{x}_{t_0}$ with Gaussian noise, the first iteration is given by

$$\begin{aligned}
\boldsymbol{x}_{t_1} &= \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t_0}) + e^{-h_1}(\boldsymbol{x}_{t_0} - \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t_0})) && \text{(CFG)} \\
\boldsymbol{x}_{t_1} &= \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\lambda}(\boldsymbol{x}_{t_0}) + e^{-h_1}(\boldsymbol{x}_{t_0} - \hat{\boldsymbol{x}}_{\varnothing}(\boldsymbol{x}_{t_0})) && \text{(CFG++)}
\end{aligned}$$

The following iterations by using the standard CFG reads

$$\boldsymbol{D}_i = \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t_{i-1}}) + \frac{1}{2r_i}\left(\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t_{i-1}}) - \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t_{i-2}})\right), \tag{A.64}$$

$$\boldsymbol{x}_{t_i} = e^{-h_i}\boldsymbol{x}_{t_{i-1}} - (e^{-h_i} - 1)\boldsymbol{D}_i. \tag{A.65}$$

Rearranging Eq. (A.64), Eq. (A.65), we can rewrite the update steps as

$$\boldsymbol{x}_{t_i} = \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t_{i-1}}) - e^{-h_i}\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t_{i-1}}) + \frac{1 - e^{-h_i}}{2r_i}\left(\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t_{i-1}}) - \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t_{i-2}})\right) + e^{-h_i}\boldsymbol{x}_{t_{i-1}}. \tag{A.66}$$

Notice that in order to apply CFG++ to Eq. (A.66), we should only keep the first term as the conditional Tweedie, and use the unconditional estimates for the rest of the components. i.e.

$$\boldsymbol{x}_{t_i} = \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\lambda}(\boldsymbol{x}_{t_{i-1}}) - e^{-h_i}\hat{\boldsymbol{x}}_{\varnothing}(\boldsymbol{x}_{t_{i-1}}) + \frac{1 - e^{-h_i}}{2r_i}\left(\hat{\boldsymbol{x}}_{\varnothing}(\boldsymbol{x}_{t_{i-1}}) - \hat{\boldsymbol{x}}_{\varnothing}(\boldsymbol{x}_{t_{i-2}})\right) + e^{-h_i}\boldsymbol{x}_{t_{i-1}} \tag{A.67}$$

DPM-solver++ 2S [111] With the same choices of σ_t, h_i, we additionally define the timesteps $\{s_i\}_{i=1}^{M}$ with $t_i > s_{i+1} > t_{i+1}$. Further, let $r_i = \frac{s_i - t_{i-1}}{t_i - t_{i-1}}$. Using standard CFG, the iteration reads

$$\boldsymbol{u}_i = e^{-r_i h_i}\boldsymbol{x}_{t_{i-1}} + (1 - e^{-r_i h_i})\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t_{i-1}}) \tag{A.68}$$

$$\boldsymbol{x}_{t_i} = \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t_{i-1}}) - e^{-h_i}\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t_{i-1}}) + \frac{1 - e^{-h_i}}{2r_i}\left(\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{u}_i) - \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t_{i-1}})\right) + e^{-h_i}\boldsymbol{x}_{t_{i-1}} \tag{A.69}$$

Applying the general transition rule from CFG to CFG++ as introduced in Eq. (6.29), we can keep all of the Tweedie estimates to be unconditional, and only change the first term of Eq. (A.69), i.e.

$$\boldsymbol{u}_i = e^{-r_i h_i}\boldsymbol{x}_{t_{i-1}} + (1 - e^{-r_i h_i})\hat{\boldsymbol{x}}_{\varnothing}(\boldsymbol{x}_{t_{i-1}}) \tag{A.70}$$

$$\boldsymbol{x}_{t_i} = \hat{\boldsymbol{x}}_{\varnothing}(\boldsymbol{x}_{t_{i-1}}) - e^{-h_i}\hat{\boldsymbol{x}}_{\varnothing}(\boldsymbol{x}_{t_{i-1}}) + \frac{1 - e^{-h_i}}{2r_i}\left(\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\lambda}(\boldsymbol{u}_i) - \hat{\boldsymbol{x}}_{\varnothing}(\boldsymbol{x}_{t_{i-1}})\right) + e^{-h_i}\boldsymbol{x}_{t_{i-1}}. \tag{A.71}$$

For the ancestral version of DPM-solver++ 2S, we can follow the general transition rule, as was shown for Euler $\rightarrow$ Euler Ancestral.

A.4 Evolution of the posterior mean: CFG vs. CFG++

Recall that one can equivalently view the evolution of the posterior mean $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$ rather than the noisy variables $\boldsymbol{x}_t$. In this context, we derive the sequential evolution of conditional posterior mean $\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}$ and $\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\lambda}$ through time t to further understand the underlying behavior of the proposed sampling.

Proposition 5. *Let $d\boldsymbol{z}(\boldsymbol{x}_t) := \boldsymbol{z}(\boldsymbol{x}_t) - \boldsymbol{z}(\boldsymbol{x}_{t+1})$ denote the discrete time evolution of some random variable $\boldsymbol{z}$ at*

time t. Then, the evolution of $\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}$ of CFG and $\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\lambda}$ of CFG++ is given by

$$d\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_t) = \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}} d\hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t) + \omega(\Delta(\boldsymbol{x}_t, \boldsymbol{c}) - \Delta(\boldsymbol{x}_{t+1}, \boldsymbol{c})) \tag{A.72}$$

$$d\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\lambda}(\boldsymbol{x}_t) = \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}} \underbrace{d\hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t)}_{\textit{uncond. shift}} + \lambda \underbrace{\Delta(\boldsymbol{x}_t, \boldsymbol{c})}_{\textit{cond. shift}}, \tag{A.73}$$

where $\Delta(\boldsymbol{x}_t, \boldsymbol{c}) := \hat{\boldsymbol{x}}_{\boldsymbol{c}}(\boldsymbol{x}_t) - \hat{\boldsymbol{x}}_{\varnothing}(\boldsymbol{x}_t)$.

Proof. We start by writing the iteration from $t+1 \to t$

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t+1}) + \sqrt{1+\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t+1}). \tag{A.74}$$

The Tweedie estimate for the next step is then written as

$$\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_t) = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}} \tag{A.75}$$

$$= \frac{\sqrt{\bar{\alpha}_t}\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t+1}) + \sqrt{1-\bar{\alpha}_t}(\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t+1}) - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_t))}{\sqrt{\bar{\alpha}_t}} \tag{A.76}$$

$$= \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_{t+1}) + \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\Big[\hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_{t+1}) - \hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t) + \omega(\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_{t+1}) - \hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_{t+1})) - \omega(\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_t) - \hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t))\Big]. \tag{A.77}$$

Using the relation $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_t) = -(\sqrt{\bar{\alpha}_t}\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_t) - \boldsymbol{x}_t)/\sqrt{1-\bar{\alpha}_t}$, we have

$$d\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_t) = \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}} d\hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t) + \omega\left(\hat{\boldsymbol{x}}_{\boldsymbol{c}}(\boldsymbol{x}_t) - \hat{\boldsymbol{x}}_{\varnothing}(\boldsymbol{x}_t)\right) - \omega\left(\hat{\boldsymbol{x}}_{\boldsymbol{c}}(\boldsymbol{x}_{t+1}) - \hat{\boldsymbol{x}}_{\varnothing}(\boldsymbol{x}_{t+1})\right). \tag{A.78}$$

Similarly, for CFG++,

$$\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\lambda}(\boldsymbol{x}_t) = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}^{\lambda}(\boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}} \tag{A.79}$$

$$= \frac{\sqrt{\bar{\alpha}_t}\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\lambda}(\boldsymbol{x}_{t+1}) + \sqrt{1-\bar{\alpha}_t}(\hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_{t+1}) - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}^{\lambda}(\boldsymbol{x}_t))}{\sqrt{\bar{\alpha}_t}} \tag{A.80}$$

$$= \hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\lambda}(\boldsymbol{x}_{t+1}) + \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\left[\hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_{t+1}) - \hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t) - \lambda(\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_t) - \hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t))\right]. \tag{A.81}$$

Hence,

$$d\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\lambda}(\boldsymbol{x}_t) = \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}} d\hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t) + \lambda(\hat{\boldsymbol{x}}_{\boldsymbol{c}}(\boldsymbol{x}_t) - \hat{\boldsymbol{x}}_{\varnothing}(\boldsymbol{x}_t)), \tag{A.82}$$

□

Proposition 5 implies that the CFG++ update of $\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\lambda}$ is decomposed into two shift terms: 1) $d\hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t)$ represents the difference between consecutive unconditional scores (i.e. unconditional shift), and 2) $\Delta(\boldsymbol{x}_t, \boldsymbol{c})$ denotes the direction of conditional guidance at time t (i.e. conditional shift). The conditional shift term is multiplied by a small interpolation factor λ in the case of CFG++, inducing a small nudge toward the condition.

CFG $\hat{\boldsymbol{x}}_{\boldsymbol{c}}^{\omega}$, on the other hand, has the same unconditional shift, but has an **oscillatory behavior** for the conditional shift. The difference between CFG/CFG++ sampling arises from the unexpected additional shift

from the previous timestep $t+1$ that exists for the CFG decomposition: $-\Delta(\boldsymbol{x}_{t+1}, \boldsymbol{c})$, and the scaling constant ω.The initial conditional shift $\omega\Delta(\boldsymbol{x}_t, \boldsymbol{c})$ with a large ω pushes the trajectory off the manifold, but cancels some of its effects by subtracting the conditional shift from the previous step $\omega\Delta(\boldsymbol{x}_{t+1}, \boldsymbol{c})$. The compounded vector $\Delta(\boldsymbol{x}_t, \boldsymbol{c}) - \Delta(\boldsymbol{x}_{t+1}, \boldsymbol{c})$ does induce a nudge closer to the condition but requires a large value of ω to have a meaningful effect, and thus is hard to interpret.

Acknowledgment

Embarking on graduate school wasn't a deeply calculated decision. It felt like the natural next step—I knew I didn't know enough, and I wanted to explore the intersection of biomedical imaging and AI. Looking back, these past six years have been transformative. I began with only a surface-level grasp of what it meant to be a "scientist." Now, I understand what it means to do science and how to be good at it. But more importantly, I now know that I love being a scientist. For this, I am immensely grateful to my advisor, Prof. Jong Chul Ye, who gave an eager but naive undergraduate the chance to learn what research is. Working with him was like embarking on an adventure with a giant; his mentorship provided a sense of stability that allowed me to grow. In retrospect, each project I completed feels like a natural progression of the last, though it rarely felt that way in real time—many describe Ph.D. studies as graduate descent, but for me, it was closer to Langevin dynamics with a very large step size.

Today, I am thrilled to officially hold a doctor in Philosophy. I would like to extend my sincere gratitude to my defense committee members—Prof. Se Young Chun, Prof. Jinwoo Shin, Prof. Juho Lee, and Prof. Minhyuk Sung—for their guidance and invaluable feedback. I am also grateful to Prof. Eun Sun Lee for her tremendous support and mentorship during her sabbatical at KAIST. Throughout my journey, I was fortunate to have mentors beyond my advisor. I thank Mike for teaching me how to think from first principles and approach problems with patience. I also thank Mauricio for imparting the power of impactful ideas and the beauty of simplicity. This journey would not have been possible without my colleagues at BISPL, who served as my teachers, collaborators, and friends. I am especially grateful to my coauthors Byeongsu, Jeongsol, and Geon Yeong, whose expertise taught me so much, and whose character I deeply admire. I feel privileged to have worked alongside my junior colleagues Sehui, Suhyeon, Jinho, Hyelin, Dohun, and Abbas. Their enthusiasm and fresh perspectives continuously inspired me and propelled my work forward.

Lastly, I want to thank my parents, from whom I have inherited my scientific inclinations. Their unwavering support made this journey possible, and only as I grow older do I fully appreciate the sacrifices they made for me. Though this journey has been challenging, it has also been deeply rewarding. I am grateful for every step along the way and for the people who made it meaningful. As I begin the next chapter, I carry forward their lessons, hoping to make an impact that honors their support and guidance. Much like when I started my Ph.D., I have only begun to grasp the meaning of holding this degree. Six years from now, I hope I understand and appreciate it deeply.

Curriculum Vitae

Name : Hyungjin Chung

E-mail : hj.chung@kaist.ac.kr

Educations

2021. 3. – 2025. 2. Ph.D., Bio & Brain engineering, **KAIST**

2019. 3. – 2021. 2. M.S., Bio & Brain engineering, **KAIST**

2015. 3. – 2019. 2. B.S., Biomedical engineering, **Korea University**

Work Experience

2024. 08. – Present Research Advisor, **EverEx**

2023. 11. – 2024. 01. Research Intern, **NVIDIA Research**

2023. 07. – 2023. 10. Student Researcher, **Google Research**

2023. 07. – 2023. 12. Research Advisor, **Team Learners**

2023. 03. – 2023. 12. Technical Writer, **Alphasignal**

2022. 06. – 2022. 08. Research Intern, **Los Alamos National Laboratory**

Professional Service

2021. 05. – Present Advisory board member, **SNUHRad-AICON**

Awards

1. **Google Conference Scholarship**, ICML 2024 2024. 05.
2. **30$^{\text{th}}$ Samsung Humantech Gold Award** (1st prize in signal processing) 2024. 02.
3. **Bronze Prize, IPIU 2024** 2024. 02.
4. **29$^{\text{th}}$ Samsung Humantech Gold Award** (1st prize in signal processing) 2023. 02.
5. **2020-2023 BISPL Best Researcher Award** 2020-2023. 12.

Publications

1. Sehui Kim*, **Hyungjin Chung***, Se Hie Park, Eui-Sang Chung, Kayoung Yi, Jong Chul Ye, "Fundus image enhancement through direct diffusion bridges", *IEEE JBHI 2024*
2. **Hyungjin Chung**, Jong Chul Ye, "Deep Diffusion Image Prior for Efficient OOD Adaptation in 3D Inverse Problems", *ECCV 2024*

3. **Hyungjin Chung**, Jong Chul Ye, Peyman Milanfar, Mauricio Delbracio, "Prompt-tuning Latent Diffusion Models for Inverse Problems", *ICML 2024*
4. **Hyungjin Chung**, Suhyeon Lee, Jong Chul Ye, "Decomposed Diffusion Sampler for Accelerating Large-Scale Inverse Problems", *ICLR 2024*
5. **Hyungjin Chung**, Hyelin Nam, Jong Chul Ye, "Review of diffusion models: theory and applications", *JKSIAM, 2024*
6. Se Hie Park, **Hyungjin Chung**, Jong Chul Ye, Kayoung Yi, "Dehazing Algorithm for Enhancing Fundus Photographs Using Dark Channel and Bright Channel Prior", *Journal of the Korean Ophthalmological Society, 2024*
7. **Hyungjin Chung**, Jeongsol Kim, Jong Chul Ye, "Direct Diffusion Bridge using Data Consistency for Inverse Problems", *NeurIPS 2023*
8. Suhyeon Lee*, **Hyungjin Chung***, Minyoung Park, Jonghyuk Park, Wi-Sun Ryu, Jong Chul Ye, "Improving 3D Imaging with Pre-Trained Perpendicular 2D Diffusion Models", *ICCV 2023*
9. Michael T. Mccann, **Hyungjin Chung**, Jong Chul Ye, Marc L. Klasky, "Score-based Diffusion Models for Bayesian Image Reconstruction", *ICIP 2023*
10. **Hyungjin Chung***, Jeongsol Kim*, Sehui Kim, Jong Chul Ye, "Parallel Diffusion Models of Operator and Image for Blind Inverse Problems", *CVPR 2023*
11. **Hyungjin Chung***, Jeongsol Kim*, Michael T. Mccann, Marc L. Klasky, Jong Chul Ye, "Diffusion Posterior Sampling for General Noisy Inverse Problems", *ICLR 2023 (spotlight)*
12. **Hyungjin Chung***, Byeongsu Sim*, Dohoon Ryu, Jong Chul Ye, "Improving Diffusion Models for Inverse Problems using Manifold Constraints", *NeurIPS 2022*
13. **Hyungjin Chung**, Byeongsu Sim, and Jong Chul Ye, "Come-Closer-Diffuse-Faster: Accelerating Conditional Diffusion Models for Inverse Problems through Stochastic Contraction", *CVPR 2022*
14. **Hyungjin Chung**, Eun Sun Lee, Jong Chul Ye, "MR Image Denoising and Super-Resolution Using Regularized Reverse Diffusion" *IEEE Transactions on Medical Imaging* 42.4 (2022): 922-934.
15. Eunju Cha*, **Hyungjin Chung***, Jaeduck Jang, Junho Lee, Eunha Lee, Jong Chul Ye, "Low-dose sparse-view HAADF-STEM-EDX tomography of nanocrystals using unsupervised deep learning", *ACS nano* 16.7 (2022): 10314-10326.
16. **Hyungjin Chung** and Jong Chul Ye, "Score-based diffusion models for accelerated MRI" *Medical image analysis* 80 (2022): 102479.
17. Mehmet Akçakaya, Burhaneddin Yaman, **Hyungjin Chung**, Jong Chul Ye, "Unsupervised Deep Learning Methods for Biological Image Reconstruction and Enhancement", *IEEE Signal Processing Magazine* 39.2 (2022): 28-44.
18. Joon Yeul Nam*, **Hyungjin Chung***, Kyu Sung Choi*, Hyuk Lee* et al., "A Deep Learning Model for Diagnosing Gastric Mucosal Lesions Using Endoscopic Images: Development, Validation, and Method Comparison", *Gastrointestinal Endoscopy* 95.2 (2022): 258-268.
19. **Hyungjin Chung**, Jong Chul Ye, "Feature Disentanglement in generating three-dimensional structure from two-dimensional slice with sliceGAN", *Nature Machine Intelligence* 3, (2021): 861-863
20. **Hyungjin Chung***, Jaeyoung Huh*, Geon Kim, Yong Keun Park, Jong Chul Ye, "Missing Cone Artifacts Removal in ODT using Unsupervised Deep Learning in Projection Domain", *IEEE Transactions on Computational Imaging* 7 (2021): 747-758

21. **Hyungjin Chung**, Eunju Cha, Leonard Sunwoo, Jong Chul Ye, "Two-Stage Deep Learning for Accelerated 3D Time-of-Flight MRA without Matched Training Data", *Medical Image Analysis* 71 (2021): 102047
22. Yoseob Han*, Jaeduck Jang*, Eunju Cha*, Junho Lee*, **Hyungjin Chung*** et al., "Deep learning STEM-EDX tomography of nanocrystals", *Nature Machine Intelligence* 3.3 (2021): 267-274.
23. Eunju Cha, **Hyungjin Chung**, Eung Yeop Kim, Jong Chul Ye, "Unpaired training of deep learning tMRA for flexible spatio-temporal resolution", *IEEE Transactions on Medical Imaging* 40.1 (2020): 166-179
24. Gyutaek Oh, Byeongsu Sim, **Hyungjin Chung**, Leonard Sunwoo, Jong Chul Ye, "Unpaired deep learning for accelerated MRI using optimal transport driven cycleGAN", *IEEE Transactions on Computational Imaging* 6 (2020): 1285-1296

Books Chapters

1. Tolga Çukur, Mahmut Yurt, Salman Ul Hassan Dar, **Hyungjin Chung**, Jong Chul Ye, "Chapter 12: Image Synthesis in Multi-Contrast MRI with Generative Adversarial Networks" *Deep Learning for Biomedical Image Reconstruction*

Preprints

1. Jinho Chang, **Hyungjin Chung**, Jong Chul Ye, "Contrastive CFG: Improving CFG in Diffusion Models by Contrasting Positive and Negative Prompts"
2. Junho Kim, **Hyungjin Chung**, Byung-Hoon Kim, "CapeLLM: Support-Free Category-Agnostic Pose Estimation with Multimodal Large Language Models"
3. Won Jun Kim*, **Hyungjin Chung***, Jemin Kim*, Byeongsu Sim, Sangmin Lee, Jong Chul Ye, "Derivative-Free Diffusion Manifold-Constrained Gradient for Unified XAI"
4. **Hyungjin Chung***, Dohun Lee*, Jong Chul Ye, "ACDC: Autoregressive coherent multimodal generation using diffusion correction"
5. Giannis Daras, **Hyungjin Chung**, Chieh-Hsin Lai, Yuki Mitsufuji, Jong Chul Ye, Peyman Milanfar, Alexandros G Dimakis, Mauricio Delbracio, "A survey on diffusion models for inverse problems"
6. **Hyungjin Chung***, Jeongsol Kim*, Geon-Yeong Park*, Hyelin Nam*, Jong Chul Ye, "CFG++: Manifold-constrained Classifier Free Guidance for Diffusion Models"
7. Abbas Mammadov*, **Hyungjin Chung***, Jong Chul Ye, "Amortized Posterior Sampling with Diffusion Prior Distillation"
8. Jeongsol Kim*, Geon-Yeong Park*, **Hyungjin Chung**, Jong Chul Ye, "Regularization by texts for latent diffusion inverse solvers"
9. Riccardo Barbano*, Alexander Denker*, **Hyungjin Chung***, Tae Hoon Roh, Simon Arrdige, Peter Maass, Bangti Jin, Jong Chul Ye, "Steerable Conditional Diffusion for Out-of-Distribution Adaptation in Imaging Inverse Problems"
10. Pinaya et al., "Generative AI for Medical Imaging: extending the MONAI Framework"

Invited talks & Lectures

1. **Texts in inverse problem solving using diffusion models**
   University of Michigan 2024. 10.

2. **Tutorial on Denoising Diffusion Model: Fundamentals & Applications**
   IEIE: Winter School on Biomedical Signal Processing 2024. 02.

3. **Adapting diffusion models for inverse problems**
   UCLA, Caltech: Grundfest Memorial Lecture Series in Graphics and Imaging 2024.0
   2023 NeurIPS Workshop on diffusion models 2023.12
   Google Research 2023.10

4. **Advances in diffusion models and their applications to inverse problems**
   Guest Lecture, Korea University 2023.11

5. Generative (diffusion) models for medical imaging
   International Congress on Magnetic Resonance Imaging (ICMRI) 2023 2023.11
   Michigan State University 2023.09
   Stanford MedAI 2023.08
   MGH, School of Medicine, Harvard University 2023.08
   BRIC academic webinar 2023.03
   45^{th} meeting, The Korean Society of Abdominal Radiology, 2022 2022.06

6. **Diffusion models: foundations and applications in biomedical imaging**
   IEEE International Symposium on Biomedical Imaging (ISBI) 2023 2023.05

7. **Diffusion models for inverse problems**
   LANL 2024.09
   Korea University, Krafton AI 2024.09
   DRGem, LG AI Research 2024.08
   TwelveLabs 2024.06
   AI SEOUL 2024 2024.02
   Inference & control group seminar, Donders Institute, Radboud Univ. 2023.01
   LANL T-CNLS seminar 2022.08

Teaching

1. **Head TA, KAIST**
   AI 618: Generative models and unsupervised learning 2024-01
   BiS 800: Machine Learning for Medical Image Analysis 2021-2

2. **TA, KAIST**
   AI 618: Generative models and unsupervised learning 2022-02
   MAS 480: Advanced Intelligence 2021-1
   BiS 452: Biomedical Imaging 2020-2
   BiS 301: Bioengineering Laboratory I 2019, 2020-1

Patents

1. "Score-based Diffusion Model for Accelerated MRI and Apparatus thereof", *US patent application*, 2023

2. “Tomography image processing method using neural network based on unsupervised learning to remove missing cone artifacts and apparatus therefor”, *Korea patent publication*, 2023
3. “Two-Stage unsupervised learning method for 3D Time-of-flight MRA reconstruction and the apparatus thereof”, *Korea patent publication*, 2023
4. “Accelerating method of conditional diffusion models for inverse problems using stochastic contraction and the apparatus thereof”, *Korea patent application*, 2021
5. “Extreme condition reconstruction method HAADF-STEM-EDX tomography using unsupervised deep learning and the apparatus thereof”, *Korea patent application*, 2021

Reviewer

1. **Conference**
   ICLR 2024-2025, NeurIPS 2022-2024, NeurIPS D&B 2023-2024, CVPR 2023-2025, ICCV/ECCV 2022-2024, MICCAI 2022-2023
2. **Journal**
   NEJM AI, Nature Communications, Medical Image Analysis
   IEEE TMI (**Gold Distinguished reviewer 2024**, **Bronze Distinguished reviewer 2023**), TPAMI, TCI, TSP, TIP, SPS