Solving Diffusion Inverse Problems with Restart Posterior Sampling

Bilal Ahmed and Joseph G. Makin

Elmore School of Electrical and Computer Engineering
Purdue University, West Lafayette IN, USA
{ahmedb, jgmakin}@purdue.edu
https://bilalhsp.github.io/RePS/

Abstract. Inverse problems—inferring an underlying signal or state from incomplete or noisy measurements—are fundamental to science and engineering. Recent approaches employ diffusion models as powerful implicit priors for such problems, owing to their ability to capture complex data distributions. However, existing diffusion-based methods for inverse problems often rely on strong approximations of the posterior distribution, require computationally expensive gradient backpropagation through the score network, or are restricted to linear measurement models.

In this work, we propose Restart for Posterior Sampling (RePS), a general and efficient framework for solving both linear and non-linear inverse problems using pre-trained diffusion models. RePS builds on the idea of restart-based sampling, previously shown to improve sample quality in unconditional diffusion, and extends it to posterior inference. Our method employs a conditioned ODE applicable to any differentiable measurement model and introduces a simplified restart strategy that contracts accumulated approximation errors during sampling. Unlike some of the prior approaches, RePS avoids backpropagation through the score network, substantially reducing computational cost. We demonstrate that RePS achieves faster convergence and superior reconstruction quality compared to existing diffusion-based baselines across a range of inverse problems, including both linear and non-linear settings.

Keywords: Posterior sampling · Diffusion priors · Image restoration

1 Introduction

Inverse problems are ubiquitous in science and engineering, arising in medical imaging, computational photography, and numerous other domains where the goal is to infer an underlying state from indirect or incomplete measurements. In most scientific applications, we have access only to partial observations that are related to the true system through a forward measurement process. The challenge lies in recovering the original signal or state from these incomplete or degraded measurements.

Examples of such settings abound: neural spiking activity reflects underlying perceptual or motor states; the number of photons captured by an image sensor corresponds to a physical scene we seek to reconstruct; MRI measurements encode the tissue and structural properties of the human body. Because the observations are incomplete, degraded, or corrupted by measurement noise, the recovery problem, *i.e.*, estimating the underlying signal from the observations, is ill-posed and requires incorporating prior knowledge about the solution space. In a probabilistic framework, the measurement process is modeled by a likelihood function, $p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$, and the constraint by a prior distribution, $p(\boldsymbol{x};\boldsymbol{\theta})$. Solving the inverse problem amounts to computing the posterior distribution, $p(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\theta})$. (Here $\boldsymbol{\theta}$ denotes the parameters of the measurement and/or prior models.)

For practical problems where the solution lies in a complex, high-dimensional space, specifying an explicit prior distribution is nontrivial. Historically, simple analytical priors, such as smoothness constraints (*e.g.*, total-variation regularization or ℓ_2 penalties), were employed to regularize the solution. With the advent of data-driven methods, however, learned models have become a powerful alternative for capturing more realistic image or signal priors. For instance, in computational imaging, Venkatakrishnan and colleagues [31] introduced the idea of replacing the proximal operator in iterative reconstruction with a learned denoiser. Following this, many groups [2, 3, 10, 11, 23, 28] have attempted to regularize inverse problems with *implicit* priors based on latent-variable generative models, like variational autoencoders (VAEs), generative adversarial networks (GANs), and diffusion models.

Diffusion models have achieved remarkable success as generative models for complex, high-dimensional, data distributions [8,9,15,20–22,24,29] such as high-resolution images, audio, and protein structures, and are now being explored for discrete domains as well. Their state-of-the-art sample quality, coupled with stable training dynamics, makes them highly attractive as priors for solving inverse problems. Nevertheless, there is a major obstacle to integrating diffusion priors with measurement models: The variables ($\boldsymbol{x}_0$) that are explicitly constrained by the measurement model (the likelihood function) are only implicitly constrained by the diffusion model (the prior)—either in the form of samples or by way of numerical integration, both computationally expensive.

One way to circumvent this obstacle is simply to introduce a third model. For example, we might model the posterior directly by training a conditional diffusion model, *e.g.* as in classifier-free guidance. Alternatively, for the cases where the likelihood function is a learned model (*e.g.*, an image classifier for the task of class-conditioned generation), if we first train an auxiliary image classifier $p(\boldsymbol{y}|\boldsymbol{x}_t;\boldsymbol{\theta})$ on pairs of *noisy* images $\boldsymbol{x}_t$ and their labels $\boldsymbol{y}$, the likelihood can be incorporated into *every* step of the reverse-diffusion process [5]. However, although effective, training a separate model for every task is inefficient, and in some cases highly impractical. Here, then, we are interested only in the "plug-and-play" framework that uses a pre-trained, unconditional diffusion model and adapts it to arbitrary inverse problems (on the same dataset).

Crudely, there are two basic approaches. The first is to separate the optimization scheme from the sampling process, and apply the constraint only in $\boldsymbol{x}_0$ space. For example, after sampling from a(n unconditioned) diffusion model, the measurement constraint can be imposed in $\boldsymbol{x}_0$ space with Langevin dynamics [36] or gradient descent [18]. This avoids backpropagation through the score network, but risks trapping the solution in a bad local mode. To address this, these approaches then add noise to the sample, pushing it out of the local mode. The entire process is then repeated, with steadily decreasing noise. This approach has been shown to recover high-quality samples. Nevertheless, it is intuitively inefficient, since the reverse diffusions (*e.g.*, via numerical integration of the ODE or SDE) are blind to the constraint.

The second basic approach is to attempt to optimize the likelihood and the prior simultaneously. Since their spaces of operation are connected through the reverse-diffusion neural network, most techniques in this category backpropagate gradients through it. For example, Graikos *et al.* [7] propose a variational-inference scheme, under which the optimization is carried out in $\boldsymbol{x}_0$ space, but which is applied to the diffusion model by making a round trip backwards through the reverse diffusion process and thence through the forward diffusion. In DPS [1], the measurement operator is "lifted" to noisy samples through the diffusion network, and again optimization requires backpropagating through it.

In this study, we propose an algorithm of the second kind (optimizing likelihood and prior simultaneously), but without backpropagating through the diffusion network. In particular, we exploit the fact that the popular DDIM sampler for diffusion models [25] can be used to express a single step of an ODE simulation—from $\boldsymbol{x}_t$ to $\boldsymbol{x}_{t-1}$—in terms of the posterior mode, $\mathbb{E}[\boldsymbol{X}_0|\boldsymbol{x}_t, \boldsymbol{y}]$. The optimal $\boldsymbol{x}_0$ is the one that lies close to the unconditional mode $\mathbb{E}[\boldsymbol{X}_0|\boldsymbol{x}_t]$ but also respects the measurement. Finding this optimal value (by *e.g.* gradient descent) does not require backpropagating through the diffusion network because it answers the conditional question, "What image $\boldsymbol{x}_0$ looks most like a denoised version of the current sample $\boldsymbol{x}_t$, subject to the constraint?" rather than the unconditional question, "What sample $\boldsymbol{x}_t$ is most congruent with this denoiser, subject to the constraint?"

Still, the method requires approximations. Here, then, we import some observations from a recent investigation of the quality of *unconditioned* diffusion sampling under various numerical-integration schemes. In particular, Xu *et al.* [35] showed that, when operating with a limited budget of neural function evaluations (NFEs), ODEs typically outperform SDEs, because they suffer from smaller discretization errors. On the other hand, given a large budget of NFEs, SDEs outperform ODEs, because they accumulate approximation errors—*e.g.*, from an imperfect score function—more slowly. Crucially, however, [35] also showed how to achieve the best of both worlds, by numerically integrating an ODE (for small discretization error), but periodically "restarting" it, by adding a large amount of noise, in order to obliterate approximation error.

Here, we propose a general framework, "restart posterior sampling" (RePS), that extends the restart sampling concept to posterior inference. We apply this

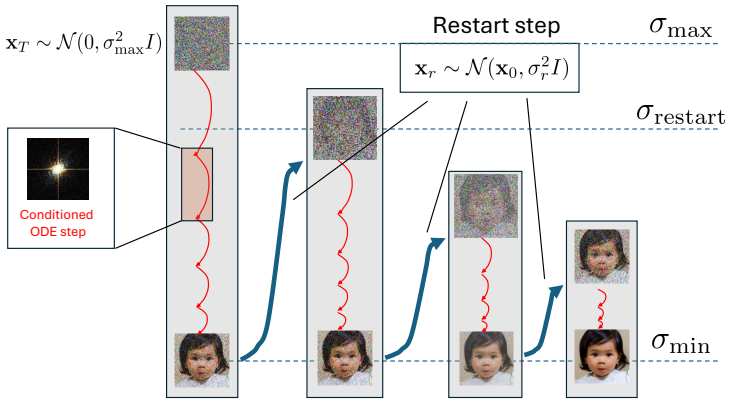

Fig. 1: Conceptual schematic for RePS. The process begins by sampling from $\mathcal{N}(\mathbf{0},\, \sigma(T)^2\mathbf{I})$ and solving the measurement-conditioned ODE (steps indicated by red arcs). Once the sample reaches the noise level corresponding to $\sigma(0)$, the sampler is "restarted" by adding noise that returns it to $\sigma(r)$. This procedure is repeated while annealing down the restart noise level.

framework to DDIM-based samplers to solve inverse problems. The key contributions of our work are:

- a unified view of recent diffusion-based plug-and-play methods under the framework of restart posterior sampling;
- an algorithm for solving such problems that avoids backpropagating through the score network [17,37], but also extends to any differentiable measurement operator, whether linear or non-linear, and allows an arbitrary number of conditioned ODE steps per restart;
- a simplified restart strategy that triggers restarts only when the sample reaches the noise-free regime: sampling is periodically restarted from σ_0 to a higher noise level σ_r, with σ_r gradually annealed from a tuned maximum value down to $\sigma_{\min}$;
- faster convergence and improved sample quality compared to prior state-of-the-art methods.

2 Background

2.1 Diffusion Models

Generative diffusion models are defined in terms of a stochastic process that transforms a complicated data distribution $p_{\text{data}}(\boldsymbol{x}_0)$ into structureless noise $p(\boldsymbol{x}_T)$. This process can be described by a stochastic differential equation (SDE), perhaps the most common of which is the "variance-exploding" version:

$$\mathrm{d}\boldsymbol{X}_t = \sqrt{2\dot{\sigma}(t)\sigma(t)}\,\mathrm{d}\boldsymbol{W}_t, \tag{1}$$

where $\sigma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a predefined noise schedule and $\boldsymbol{W} \in \mathbb{R}^d$ is a standard Wiener process. The noise schedule is defined such that $\sigma(0) = 0$ and $\sigma(T)$ is large enough to destroy any structure in the data. It follows from Eq. (1) that the noise-perturbed distribution at step t, $p(\boldsymbol{x}_t|\boldsymbol{x}_0)$, takes the form $\mathcal{N}\big(\boldsymbol{x}_0,\, \sigma(t)^2\mathbf{I}\big)$. The goal of the generative model is to invert the forward diffusion process; *i.e.*, to start with a sample $\boldsymbol{x}_T \sim \mathcal{N}\big(\mathbf{0},\, \sigma(T)^2\mathbf{I}\big)$ and denoise it all the way back to $\boldsymbol{x}_0 \sim p_{\text{data}}(\boldsymbol{x}_0)$. Indeed, it is known that the forward diffusion process in Eq. (1) is invertible, with the marginal distribution over observations $\boldsymbol{x}_0$ given [12, 29] either by the SDE

$$\mathrm{d}\boldsymbol{X}_t = -2\dot{\sigma}(t)\sigma(t)\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{X}_t)\mathrm{d}t + \sqrt{2\dot{\sigma}(t)\sigma(t)}\mathrm{d}\bar{\boldsymbol{W}}_t, \tag{2}$$

with $\bar{\boldsymbol{W}}_t$ a reverse-time Wiener process; or by the ODE

$$\mathrm{d}\boldsymbol{X}_t = -\dot{\sigma}(t)\sigma(t)\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{X}_t)\mathrm{d}t. \tag{3}$$

Evidently, to generate samples, we need estimates of the "score," *i.e.* $\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$, at all $\boldsymbol{x}_t$. This can be achieved by training a neural network $\mathbf{s}_{\boldsymbol{\theta}}(\boldsymbol{x}, \sigma)$ to match $\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$ approximately at all noise levels $\sigma(t)$ [27, 32].

2.2 Inverse Problems

In the setting of inverse problems, the signal of interest $\boldsymbol{X}_0 \in \mathbb{R}^n$ is not directly observed; rather, we have only measurements (or observe only degraded signals), $\boldsymbol{Y} \in \mathbb{R}^m$. On the other hand, the relationship between the signal and its measurement is taken to be known or well described by a model. For additive Gaussian noise, the measurement model is defined as

$$\boldsymbol{Y} = \mathbf{h}(\boldsymbol{X}_0) + \boldsymbol{Z}, \tag{4}$$

where $\mathbf{h}(\cdot)$: $\mathbb{R}^n \to \mathbb{R}^m$ is the measurement (or degradation) operator, which is in general nonlinear, and $\boldsymbol{Z} \sim \mathcal{N}\big(\mathbf{0},\, \sigma_n^2\mathbf{I}\big)$ is the measurement noise. The aim is to "invert" this model to recover $\boldsymbol{X}_0$ from $\boldsymbol{Y}$. Such problems are often ill-posed: solutions may be non-existent, non-unique, or unstable with respect to measurement noise.

To address this, prior knowledge about the solution is imposed, typically through a Bayesian formulation. By Bayes' rule, the posterior distribution can be written as $p(\boldsymbol{x}_0|\boldsymbol{y};\boldsymbol{\theta}) \propto p(\boldsymbol{x}_0;\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{x}_0;\boldsymbol{\theta})$. Therefore, given the prior distribution, $p(\boldsymbol{x};\boldsymbol{\theta})$, and measurement model, $p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$, one can (at least in theory) compute or sample from the posterior distribution, $p(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\theta})$, to generate solutions. In particular, we can use Eq. (2) or Eq. (3) to generate samples from the posterior, with the score of the posterior distribution,

$$\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{y};\boldsymbol{\theta}) = \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t;\boldsymbol{\theta}) + \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{y}|\boldsymbol{x}_t;\boldsymbol{\theta}), \tag{5}$$

in place of the unconditional score—provided we can compute it. The unconditional score $\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t;\boldsymbol{\theta})$ is (by hypothesis) available in the form of a diffusion model. But the quantity $\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{y}|\boldsymbol{x}_t;\boldsymbol{\theta})$ relates the measurement to *noisy* observations $\boldsymbol{x}_t$, and is not generically computable in closed form from the likelihood score, $\nabla_{\boldsymbol{x}_0} \log p(\boldsymbol{y}|\boldsymbol{x}_0;\boldsymbol{\theta})$.

2.3 Related Work

Several methods proposed to use score-based diffusion models for conditional generation in a plug-and-play fashion. Most of these approaches aim to guide the diffusion sampling process by incorporating measurement constraints. If we are willing to restrict ourselves to linear problems, then the singular-value decomposition [14] or projections onto the measurement space [3] can be used to enforce the constraints. Alternatively, a popular line of research extends these approaches to non-linear problems by approximating the conditional score $p(\boldsymbol{y}|\boldsymbol{x}_t;\boldsymbol{\theta})$ in terms of the likelihood score $p(\boldsymbol{y}|\boldsymbol{x}_0;\boldsymbol{\theta})$. Methods such as DPS [1] use point estimates, while others employ simple unimodal distributions with estimated variances [26]. A major drawback of this class of methods is their computational overhead, as they require evaluating the Jacobian of the score function.

Wang *et al.* [33] reformulated the problem from optimizing directly over $\boldsymbol{x}_0$ to optimizing the noise initialization $\boldsymbol{x}_T$, using an ODE as a deterministic mapping between these spaces. However, this method involves backpropagating through multiple score network evaluations, which can be computationally expensive.

Variational inference, a standard approach when the posterior is not available in closed form, approximates the posterior distribution with a simpler family of distributions. VI has recently been applied to inverse problems with diffusion priors, achieving some high-quality results [7, 19]. However, the objective, which is computed with a sample average under the forward diffusion process, turns out to be very sensitive to which noise levels are sampled, and at which gradient step. Very careful tuning is required to achieve good results.

A very different approach is to use optimal-control strategies to guide the diffusion sampler. Li and Pereira [16] demonstrated promising results along these lines, but their experiments were limited to linear problems and required a large number of function evaluations (up to 2500 NFEs for linear tasks).

For *unconditional* diffusion sampling, Xu *et al.* [35] showed that, under high NFE regimes, SDE-based samplers outperform ODE-based samplers, as they more effectively contract accumulated approximation errors. Building on this, Li and Wang [17] demonstrated that a DDIM-like sampler can be formulated as either an ODE or SDE sampler, particularly for linear inverse problems, and showed that conditioned SDE sampling produces higher-quality samples than conditioned ODE sampling by mitigating accumulated approximation errors. We discuss the approach of Li and Wang in greater detail below.

For unconditional generation, Xu *et al.* [35] also showed that ODE-based sampling with intermittent restarts—in which large amounts of noise are added only after many steps rather than small amounts at every step as in SDE sampling—outperforms both pure ODE and pure SDE formulations. This approach combines the low discretization error of ODEs with the noise-injection benefits of SDEs, and is the inspiration for our own restart procedure. They did not, however, provide a simple restart strategy; instead, they employed restarts at multiple noise levels along the trajectory, often repeating several restarts at each level.

Algorithm 1 Restart Posterior Sampling (RePS)

Require: s_θ, $\sigma_{\max}$, σ_0, σ_{res}, $\sigma_{\min}$, N_{res}, ρ_{res}, N_{ode}, ρ_{ode}

Restart schedule: $\{\sigma_r\}_{r=1}^{K-1} \leftarrow \textsc{GetSchedule}(\sigma_{\text{res}}, \sigma_{\min}, N_{\text{res}} - 1, \rho_{\text{res}})$

$\sigma_K \leftarrow \sigma_{\max}$, Sample: $\boldsymbol{x}_K \sim \mathcal{N}(0, \sigma_K^2 I)$

for $r = K, K-1, \ldots, 1$ **do**

Step 1: Conditioned ODE - N_{ode} **steps**

ODE schedule: $\{\sigma_t\}_{t=1}^{N_{\text{ode}}} \leftarrow \textsc{GetSchedule}(\sigma_{\text{r}}, \sigma_0, N_{\text{ode}}, \rho_{\text{ode}})$

$\boldsymbol{x}_t \leftarrow \boldsymbol{x}_r$

for $t = N_{\text{ode}} \ldots 1$ **do**

$\mathbb{E}[\boldsymbol{X}_0|\boldsymbol{x}_t] \leftarrow \boldsymbol{x}_t + \sigma_t^2 s_\theta(\boldsymbol{x}_t, \sigma_t)$

$\hat{\boldsymbol{x}}_{0|y} \leftarrow \operatorname{argmin}_{\boldsymbol{x}} \left\{ \frac{1}{2}\|\boldsymbol{y} - \mathbf{h}(\boldsymbol{x})\|_2^2 + \lambda \|\boldsymbol{x} - \mathbb{E}[\boldsymbol{X}_0|\boldsymbol{x}_t]\|_2^2 \right\}$

$\boldsymbol{x}_{t-1} = \boldsymbol{x}_t + \frac{\sigma_t - \sigma_{t-1}}{\sigma_t} \left(\hat{\boldsymbol{x}}_{0|y} - \boldsymbol{x}_t\right)$

end for

Step 2: Restart

$\boldsymbol{x}_{r-1} \sim \mathcal{N}\left(\boldsymbol{x}_0, \sigma_{r-1}^2 I\right)$

end for

return $\boldsymbol{x}_0$

Recently, a successful line of work has decoupled diffusion sampling from conditioning, using an unconditional diffusion sampler followed by conditioning in the $\boldsymbol{x}_0$ space, and then propagating back to the next (lower) noise level using the forward diffusion process [18, 36]. DCDP [18] enforces the constrain in $\boldsymbol{x}_0$ space via gradient descent; whereas DAPS samples from the posterior with Langevin dynamics. DAPS, in particular, is the current state of the art [36].

Reinterpreting Recent Models in Light of RePS Like the restart diffusion sampling of Xu and colleagues [35], we can view our restart posterior sampling framework as consisting of two interleaved stages. The first stage obtains an approximate estimate of the conditioned sample $\boldsymbol{x}_{0|y}$, while the second uses the forward diffusion process to move the sample back to a higher noise level, thereby "restarting" the sampling process. This stochastic restart obliterates the accumulated approximation error. In fact, we can view several recent posterior-sampling approaches in terms of this general framework, as follows.

Li and Wang [17] were explicitly motivated by restart sampling, so their algorithm breaks naturally into these two stages. They interpreted the first stage as denoising into the pixel space with Tweedie's formula, followed by conditioning via gradient descent (in pixel space) to find a MAP estimate. Although Li and Wang interpret the two stages together as a single step of an SDE solver, Tweedie's formula by itself is equivalent to numerical integration of the reverse-diffusion *ODE* with a single Euler step (see Algorithm 1 in the Supp. Materials). DiffPIR [37] is essentially identical, except that before restarting, the MAP estimate is interpolated back towards the noisy sample. (Algorithm 2 in the Supp. Materials) Like Li & Wang, DiffPIR was demonstrated only on linear problems.

Described this way, a natural generalization of these algorithms is to break the numerical integration into multiple steps, before (as lately described) conditioning and then restarting. This corresponds to DAPS [36] and DCDP [18]—although these algorithms were not derived from this perspective. The two methods differ from each other only in their conditioning mechanism: DCDP makes a MAP estimate via gradient descent, whereas DAPS uses Langevin dynamics (see Algorithms 4 and 5 in the Supp. Materials).

Nevertheless, this is not the only possible generalization of Li & Wang and DiffPIR. An alternative is to embed the *entire* first stage, the conditioning as well as the denoising, inside a single step of an ODE solver, and then to take several such steps (Algorithm 1). The advantage is that, by directly guiding the ODE trajectory toward the desired mode, rather than first producing an unconstrained sample that may drift toward an undesirable mode, the algorithm may converge more rapidly.

The design space of (what we have called) the first stage is summarized in Fig. 2. Methods that first solve an unconditioned ODE before performing a separate conditioning step lie along one axis of this two-dimensional design space. This includes DAPS [36] and DCDP [18]. Since DiffPIR and Li & Wang numerically integrate over only a single step, the distinction between conditioning within vs. after the integration of the ODE collapses, so these algorithms, too, sit on this first axis of design space. Indeed, under this interpretation, they become a special case of DCDP, which its authors called "DCDP-Tweedie's" [18].

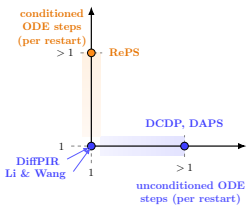

Fig. 2: Design space.

The complementary dimension, corresponding to conditioned ODEs in which denoising and conditioning are interleaved, remains largely unexplored. DiffPIR and Li & Wang can be interpreted as limiting cases (and therefore sit at the intersection of the two axes of design space). Our proposed method, RePS, explores this new dimension by performing multiple conditioned ODE steps per restart. The complete RePS algorithm is presented in Algorithm 1. We show in Sec. 4 that this does indeed typically produce faster convergence and superior reconstruction quality across a range of linear and non-linear inverse problems.

3 Methods

3.1 Measurement-Conditioned ODEs

Following Karras *et al.* [12], we use the ODE given by Eq. (3) with noise schedule defined as $\sigma(t) = t$. We can simulate this ODE using the posterior score to sample from the posterior distribution [1, 29]. Rewriting Eq. (3) in terms of $\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{y};\boldsymbol{\theta})$ to sample from $p(\boldsymbol{x}_0|\boldsymbol{y};\boldsymbol{\theta})$, we obtain

$$\mathrm{d}\boldsymbol{X}_t = -\dot{\sigma}(t)\sigma(t)\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{X}_t|\boldsymbol{y};\boldsymbol{\theta})\mathrm{d}t. \tag{6}$$

We can solve this ODE by stepping backwards in time using the Euler method:

$$\boldsymbol{x}_{t-\Delta t} = \boldsymbol{x}_t - \Delta t \frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t}, \tag{7}$$

where $\Delta t > 0$. Under our noise schedule, we have $\Delta t = \sigma(t) - \sigma(t - \Delta t)$, and the Euler update becomes

$$\boldsymbol{x}_{t-\Delta t} = \boldsymbol{x}_t + \big(\sigma(t) - \sigma(t-\Delta t)\big)\sigma(t)\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{y};\boldsymbol{\theta}). \tag{8}$$

Following a Tweedie-style argument (see Sec. 1.1 in the Supp. Materials), we obtain

$$\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{y};\boldsymbol{\theta}) = (\mathbb{E}[\boldsymbol{X}_0|\boldsymbol{x}_t, \boldsymbol{y}] - \boldsymbol{x}_t)/\sigma^2(t), \tag{9}$$

allowing us to express a measurement-conditioned numerical-integration step in terms of $\mathbb{E}[\boldsymbol{X}_0|\boldsymbol{x}_t, \boldsymbol{y}]$:

$$\boldsymbol{x}_{\,t-\Delta t} = \boldsymbol{x}_{\,t} + \frac{\big(\sigma(t) - \sigma(t-\Delta t)\big)}{\sigma(t)}(\mathbb{E}[\boldsymbol{X}_0|\boldsymbol{x}_t, \boldsymbol{y}] - \boldsymbol{x}_t). \tag{10}$$

The mean of the posterior $p(\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y};\boldsymbol{\theta})$ is generally difficult to compute, so we approximate it by the mode (MAP) of the distribution. Using the relation $p(\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y};\boldsymbol{\theta}) \propto p(\boldsymbol{y}|\boldsymbol{x}_0;\boldsymbol{\theta})p(\boldsymbol{x}_0|\boldsymbol{x}_t;\boldsymbol{\theta})$ and approximating $p(\boldsymbol{x}_0|\boldsymbol{x}_t;\boldsymbol{\theta})$ as $\mathcal{N}\big(\mathbb{E}[\boldsymbol{X}_0|\boldsymbol{x}_t],\, \sigma_t^2\mathbf{I}\big)$, the MAP estimate becomes

$$\boldsymbol{x}_{\mathrm{MAP}} = \operatorname*{argmin}_{\boldsymbol{x}} \left\{ \|\boldsymbol{y} - \mathbf{h}(\boldsymbol{x})\|_2^2 + \frac{\sigma_n^2}{\sigma_t^2}\|\boldsymbol{x} - \mathbb{E}[\boldsymbol{X}_0|\boldsymbol{x}_t]\|_2^2 \right\}. \tag{11}$$

In practice, we make $\sigma_n^2/\sigma_t^2 =: \lambda$ a tunable parameter, fixed for all time. Li and Wang [17] proceed to write the solution to the optimization in Eq. (11) in closed form for linear problems—but that restricts their ODE sampler to linear or very specific non-linear problems. To keep our method general, we solve the optimization using gradient descent at each step of the numerical integration. The resulting ODE sampler can then be used for any linear or non-linear problems with differentiable measurement operators. Since this does not involve gradient backpropagation through the score network, the computational costs remain low.

We solve the conditioned ODE defined in Eqs. (10) and (11) from $\sigma(t)$ down to σ_0 using a polynomial interpolation schedule:

$$\sigma_s = \left(\sigma_{\mathrm{start}}^{\frac{1}{\rho}} + s\left(\sigma_{\mathrm{end}}^{\frac{1}{\rho}} - \sigma_{\mathrm{start}}^{\frac{1}{\rho}}\right)\right)^{\rho}, \tag{12}$$

where $s \in [0, 1]$ is a normalized step parameter interpolating between σ_{start} (at $s = 0$) and σ_{end} (at $s = 1$). In all experiments in this work, we use $\rho_{\mathrm{ode}} = 7$ for the conditioned ODE, following common practice in diffusion-based sampling [12].

3.2 Restart Strategy

Instead of solving the conditioned ODEs given by Eqs. (10) and (11) from $\sigma_{\max}$ to σ_0 in a single pass, we adopt a restart-based approach as shown in Fig. 1. Specifically, we first solve the ODE for a small number of steps down to σ_0, then perform a forward diffusion step to reintroduce noise at a predefined restart level σ_{restart}. From this point, we again solve the ODE backward to σ_0. This process is repeated multiple times, with the restart level gradually annealed from σ_{restart} to $\sigma_{\min}$.

Since we employ the VE-SDE formulation, each restart step is defined as:

$$\boldsymbol{x}_r \sim \mathcal{N}\big(\boldsymbol{x}_0,\, \sigma_r^2 \mathbf{I}\big), \tag{13}$$

where σ_r denotes the restart noise level. Xu *et al*. [35] described a restart scheme that injected noise at multiple arbitrary levels and repeated each level several times (shown in Algorithm 3 in the Supp. Materials). In contrast, our strategy replaces that complexity with a much simpler schedule, allowing us to employ restart-based sampling as an effective alternative to standard SDE or ODE approaches.

In all our experiments, we adopt a simple restart schedule based on the very same polynomial interpolation used for the ODE simulator, namely, Eq. (12). Specifically, the restart schedule interpolates from the maximum restart level σ_{restart} to the minimum restart level $\sigma_{\min}$. We find that concentrating restart iterations at smaller noise levels improves performance. Accordingly, we tune σ_{restart} separately for each task, while keeping $\sigma_{\min}$ fixed. The full sampling procedure is summarized in Algorithm 1.

4 Experiments

We evaluate our method on two image datasets: ImageNet 256×256 [4] and FFHQ 256×256 [13]. For ImageNet, we use the pretrained diffusion model publicly released by Dhariwal and Nichol [5], and for FFHQ, we use the pretrained model provided by Chung et al. [1]. Although both of these models were trained using DDPM framework that corresponds to VP-SDE, following [12] we use the unified VE-SDE based sampler for these models. We consider a range of linear and non-linear inverse problems that are widely studied in the literature on image restoration and reconstruction. For all experiments, we begin from a noise level of $\sigma_{\max} = 100$ and use 10 steps of the conditioned ODE to move from the starting noise level (or the current restart noise level) down to $\sigma_0 = 0.01$, following the polynomial schedule defined in Eq. (12) with $\rho_{\text{ode}} = 7$. The restart schedule is then applied, beginning at task-specific σ_{restart} and interpolating down to a minimum restart level $\sigma_{\min} = 0.1$, using the same polynomial form with $\rho_{\text{restart}} = 15$. In addition to σ_{restart}, we tune the number of gradient updates per step N, the learning rate η, and the parameter λ for each task; these are summarized in Sec. 2.2 in the Supp. Materials.

Table 1: Quantitative results for FFHQ and Imagenet on a set of linear problems.

Task	Method	FFHQ			ImageNet		
		PSNR (↑)	SSIM (↑)	LPIPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)
Super Resolution 4×	RePS (ours)	**29.95**	**0.845**	**0.155**	**26.12**	**0.708**	**0.259**
	Li and Wang [17]	29.21	0.805	0.256	25.68	0.682	0.307
	DAPS (code)	29.55	0.793	0.186	25.70	0.654	0.285
	DPS	25.86	0.753	0.269	21.13	0.489	0.361
	DDRM	26.58	0.782	0.282	22.62	0.521	0.324
	DDNM	28.03	0.795	0.197	23.96	0.604	0.475
	DCDP	28.66	0.807	0.178	-	-	-
	FPS-SMC	28.42	0.813	0.204	24.82	0.703	0.313
	DiffPIR	26.64	-	0.260	23.18	-	0.371
Inpaint (Box)	RePS (ours)	24.32	**0.844**	**0.122**	20.71	**0.787**	**0.186**
	Li and Wang [17]	24.75	0.817	0.167	21.39	0.779	0.217
	DAPS (code)	**24.88**	0.755	0.174	21.40	0.721	0.228
	DPS	22.51	0.792	0.209	18.94	0.722	0.257
	DDRM	22.26	0.801	0.207	18.63	0.733	0.254
	DDNM	24.47	0.837	0.235	21.64	0.748	0.319
	DCDP	23.89	0.760	0.163	-	-	-
	FPS-SMC	24.86	0.823	0.146	**22.16**	0.726	0.208
Inpaint (Random)	RePS (ours)	32.49	0.899	0.090	28.97	0.829	0.115
	Li and Wang [17]	**34.11**	**0.926**	**0.071**	29.67	**0.861**	**0.114**
	DAPS (code)	30.94	0.807	0.155	27.72	0.742	0.177
	DPS	25.46	0.823	0.203	23.52	0.745	0.297
	DDNM	29.91	0.817	0.121	**31.16**	0.841	0.191
	DCDP	30.69	0.842	0.142	-	-	-
	FPS-SMC	28.21	0.823	0.261	24.52	0.701	0.316
Gaussian deblurring	RePS (ours)	**29.92**	**0.841**	**0.146**	26.21	**0.703**	**0.255**
	Li and Wang [17]	29.89	0.831	0.217	25.75	0.681	0.296
	DAPS (code)	29.75	0.794	0.177	26.00	0.668	0.260
	DPS	25.87	0.764	0.219	20.31	0.598	0.397
	DDRM	24.93	0.732	0.239	21.26	0.564	0.443
	DDNM	28.20	0.804	0.216	**28.06**	**0.703**	0.278
	DCDP	27.50	0.699	0.304	-	-	-
	FPS-SMC	26.54	0.773	0.253	23.91	0.601	0.387
	DiffPIR	27.36	-	0.236	22.80	-	0.355
Motion deblurring	RePS (ours)	**32.27**	**0.875**	**0.115**	**28.95**	**0.801**	**0.169**
	Li and Wang [17]	29.90	0.766	0.207	27.70	0.733	0.238
	DAPS (code)	31.80	0.843	0.135	28.87	0.781	0.171
	DPS	24.52	0.801	0.246	18.96	0.629	0.423
	DCDP	25.08	0.512	0.364	-	-	-
	FPS-SMC	27.39	0.826	0.227	24.52	0.647	0.326
	DiffPIR	26.57	-	0.255	24.01	-	0.366

Linear Problems (1) super-resolution, (2) inpainting (box mask), (3) inpainting (random mask), (4) Gaussian deblurring, and (5) motion deblurring. In super-resolution, the goal is to recover a high-resolution image from a low-resolution observation obtained by bicubic downsampling with a factor of 4. In box inpainting, each image is masked by a randomly placed box of size 128×128, while in random inpainting, 70% of the pixels are randomly masked. For Gaussian deblurring, we use a blur kernel of size 61×61 with a standard deviation of 3.0, and for motion deblurring, we use a kernel of the same size with a standard deviation of 0.5. In all tasks, Gaussian noise with a standard deviation of $\sigma_n = 0.05$ is added to the measurements.

Non-Linear Problems The non-linear inverse problems considered include: (1) phase retrieval, (2) non-linear deblurring, and (3) high dynamic range (HDR) reconstruction. In phase retrieval, the objective is to reconstruct an image from only the magnitude of its Fourier transform, where the phase information is missing. Since this is a highly ill-posed problem, we follow previous work [1, 36] in using measurements oversampled by a factor of 2, generating 4 reconstructions per measurement, and reporting the best score among them. Non-linear deblurring involves removing blur introduced by a neural network. For this task, we use the pretrained model publicly released by Tran *et al.* [30]. In HDR, the goal is to recover the full dynamic range of pixel intensities from measurements in which pixel values are compressed to a smaller range by a factor of 2. As in the linear problems, Gaussian noise with $\sigma_n = 0.05$ is added to the measurements.

Baselines and Metrics. We compare our method against the following baselines: Li and Wang [17], DAPS [36], DDRM [14], DDNM [34], DPS [1], DCDP [18], FPS-SMC [6], and DiffPIR [37] for linear problems; and DAPS [36], DPS [1], RED-diff [19], and DCDP [18] for non-linear problems. We quantitatively evaluate the methods on 100 validation images from each of FFHQ and ImageNet and report three standard metrics: peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM) and learned perceptual image patch similarity (LPIPS). These metrics assess the similarity between the generated and ground-truth images, thereby reflecting the degree of measurement consistency.

Since we use the same test setup as DAPS, most baseline results are reported from their paper. However, results obtained by running the DAPS code did not exactly match the reported values, so we report the numbers from running their official code. For comparison, we include their original results from the paper in the supplementary material. Li and Wang [17] did not provide code, so we implemented their method ourselves; we report their results for 1000 NFEs, using gradient descent rather than the closed-form solution.

4.1 Main Results

In Sec. 3 in the Supp. Materials, we present qualitative visualisations for both linear and non-linear inverse problems. These results show that RePS consistently recovers fine structural and textural details across a variety of tasks, including phase-retrieval and non-linear deblur. We also show multiple reconstructions of the same input: for phase-retrieval, RePS produces high-fidelity results reliably; for inpainting, it generates diverse yet plausible outputs in the masked regions.

Quantitative comparisons between RePS and baseline methods are shown in Tab. 1 for linear tasks and Tab. 2 for non-linear tasks. While individual baselines such as Li and Wang [17] for inpaint (random) and RED-diff for non-linear deblur achieve strong performance on specific tasks, RePS nonetheless delivers the best overall results across the full range of tasks studied. We discuss the comparison with DAPS separately later in this section, as it is the strongest baseline and achieves consistently strong performance across all tasks. RePS can

Table 2: Quantitative results for FFHQ and Imagenet on a set of nonlinear problems

Task	Method	FFHQ			ImageNet		
		PSNR (↑)	SSIM (↑)	LPIPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)
Phase-Retrieval	RePS (ours)	30.41±4.45	**0.824**±0.119	**0.152**±0.107	20.12±7.18	0.449±0.260	0.419±0.184
	DAPS (code)	**30.72**±3.15	0.809±0.081	0.157±0.073	**22.32**±6.51	**0.514**±0.219	**0.343**±0.155
	DPS	17.64±2.97	0.441±0.129	0.410±0.090	16.81±3.61	0.427±0.143	0.447±0.099
	RED-diff	15.60±4.48	0.398±0.195	0.596±0.092	14.98±3.75	0.386±0.057	0.536±0.129
	DCDP	28.65±8.09	0.781±0.217	0.203±0.196	-	-	-
Nonlinear deblur	RePS (ours)	29.02±1.76	**0.797**±0.031	0.165±0.030	27.58±3.28	0.745±0.082	**0.191**±0.056
	DAPS (code)	28.79±1.54	0.781±0.033	0.177±0.030	27.47±3.21	0.737±0.084	0.198±0.052
	DPS	23.39±2.01	0.623±0.082	0.278±0.060	22.49±3.20	0.591±0.101	0.306±0.081
	RED-diff	**30.86**±0.51	0.795±0.028	**0.160**±0.034	**30.07**±1.41	**0.754**±0.023	0.211±0.083
	DCDP	27.92±2.64	0.779±0.067	0.183±0.051	-	-	-
High dynamic range	RePS (ours)	**27.96**±3.54	**0.872**±0.080	**0.145**±0.071	26.37±4.05	**0.843**±0.117	**0.157**±0.103
	DAPS (code)	27.52±3.58	0.840±0.084	0.157±0.064	**26.50**±4.70	0.812±0.140	0.170±0.113
	DPS	22.73±6.07	0.591±0.141	0.264±0.156	19.23±2.52	0.582±0.082	0.503±0.106
	RED-diff	22.16±3.41	0.512±0.083	0.258±0.089	22.03±5.90	0.601±0.094	0.274±0.198

be easily extended to latent diffusion models. A comparison between latent and pixel-based RePS on FFHQ is provided in Sec. 3.2 in the Supp. Materials; RePS (latent) performs slightly worse than its pixel-based counterpart.

Calibration of Posterior Sampler To assess whether RePS samples from the posterior distribution rather than merely producing a point estimate, we evaluate its calibration on inherently uncertain tasks, namely box inpainting and phase retrieval. Specifically, we compute the ensemble skill, spread, and spread-to-skill ratio for both tasks, as shown in Tab 2 in the Supp. Materials. The non-trivial spread-to-skill ratios indicate that the posterior sampler does not collapse to a point estimate. In addition, the spatial spread map for box inpainting, shown in Fig. 6 in the Supp. Materials, visually demonstrates that the sampler exhibits greater uncertainty within the inpainting mask than outside it.

Comparison to SDE and ODE To highlight the performance gain achieved through restart sampling, we compare PSNR and LPIPS on Gaussian Deblur in Fig. 3a using three approaches: (i) the same ODE without any restart, (ii) an SDE derived via the Euler–Maruyama method (details in the Appendix) with identical conditioning but continuous noise injection, and (iii) our proposed RePS method. The results confirm that the ODE exhibits low discretisation error and converges rapidly at low NFEs, while the SDE delivers stronger performance at higher NFEs due to the error contraction effects of stochastic noise. RePS combines both strengths as it converges quickly and attains the best performance across the full NFE range.

As discussed in Sec. 2, Zhu *et al.* [37] and Li and Wang [17] employ an SDE derived from the DDIM sampler, which can be viewed as a special case of our RePS framework—that is, restarting interpolated with ODE simulation,

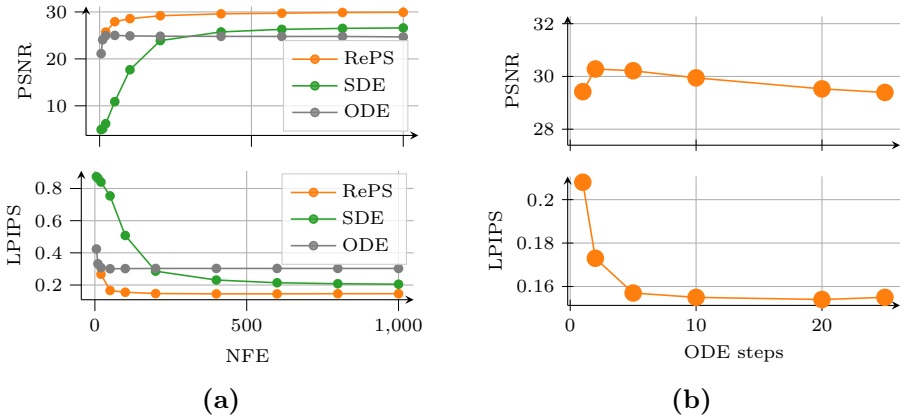

Fig. 3: Effect of type of sampler and number of steps between restarts. (a) Comparison of ODE and SDE samplers with RePS in terms of PSNR (top) and LPIPS (bottom) for Gaussian deblurring on FFHQ, as a function of NFEs. RePS achieves faster convergence and better overall performance. (b) RePS's PSNR and LPIPS on the super-resolution problem (FFHQ) for different numbers of conditioned ODE steps.

but using only a single ODE step. We present results in Fig. 3b that examine the influence of the number of ODE steps taken between successive restarts (with the total NFEs held constant at 1000) for super resolution. The leftmost data point in the figure corresponds to Eq. 15 in the Supp. Materials; *i.e.*, one conditioned ODE step between restarts. The results show that increasing the number of ODE steps between restarts improves performance up to a point. Beyond approximately 20 ODE steps, LPIPS begins to worsen, while PSNR peaks around 2 ODE steps. This result is consistent with our comparison to the standard Euler–Maruyama SDE and is also reflected in our main results in Tabs. 1 and 2, where RePS outperforms DiffPIR across all tasks and also outperforms Li and Wang [17] in all tasks except random inpainting. We provide similar figures for the remaining linear problems on FFHQ in Fig. 4 in the Supp. Materials.

Comparison to DAPS In addition to the results presented in Tabs. 1 and 2, we compare the distributions of LPIPS scores for RePS and DAPS in Fig. 5 in the Supp. Materials. On FFHQ, RePS significantly outperforms DAPS on all tasks. On ImageNet, RePS outperforms DAPS on four tasks, is competitive on three, and underperforms on one. Phase retrieval on ImageNet is the only task for which DAPS substantially outperforms RePS.

We also compare RePS with DAPS across different numbers of function evaluations (NFEs) in Fig. 4a. RePS achieves better PSNR and LPIPS than DAPS across all NFEs on the motion deblurring task. For completeness, we provide results including SSIM scores for all linear and nonlinear problems in Sec. 3 in the Supp. Materials. In Fig. 4b we compare the per-image sampling time of RePS

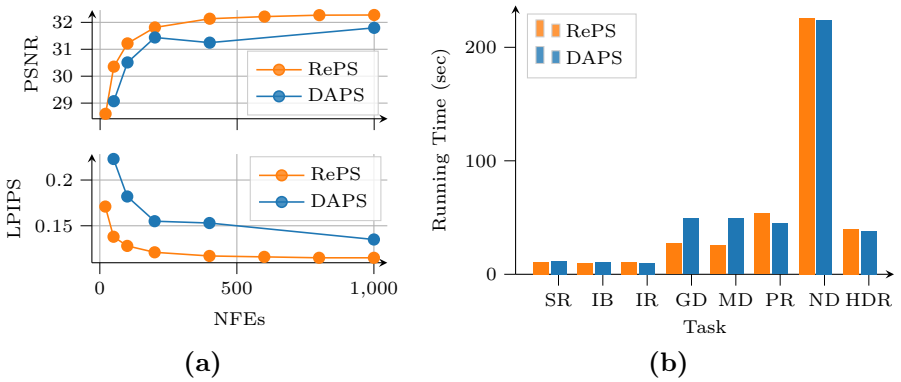

Fig. 4: Sample quality and sampling times: RePS vs. DAPS. (a) PSNR and LPIPS on FFHQ motion deblurring as a function of NFEs. RePS achieves better performance across all NFEs. (b) Time per sample (in seconds) for all tasks: super-resolution (SR), inpaint box (IB) and random (IR), Gaussian deblur (GD), motion deblur (MD), phase retrieval (PR), nonlinear-deblur (NB), and high dynamic range (HDR).

and DAPS across all tasks. The times shown correspond to the maximum NFEs used for each task (*i.e.*, 1k for linear problems and 4k for nonlinear problems). Overall, the sampling times are comparable: RePS is slightly faster on linear tasks and slightly slower on nonlinear tasks, but the differences are small.

5 Conclusions

Score-based ODE samplers benefit from low discretization error, enabling fast convergence and high-quality reconstructions, while SDE samplers inject stochasticity at each step to contract accumulated errors. Restart sampling combines both advantages by interleaving short ODE trajectories with large noise injections, producing higher-quality samples.

We extend this concept to posterior sampling in RePS by running multiple conditioned ODE steps interleaved with restart operations. We propose a simple and task-agnostic restart strategy that is easy to implement. This design achieves both faster convergence and enhanced sample fidelity compared to existing baselines. Our experiments show that RePS consistently produces better-quality samples across a diverse set of tasks, demonstrating that the restart framework offers an efficient and effective way to improve sampling quality.

Acknowledgments

We thank the anonymous reviewer (mJW4) for suggesting the interpretation of the first stage of the restart framework through the lens of a two-dimensional design space. This work was supported by NSF CAREER 2339781 and NIH R01 (1R01DC021600-01).

References

1. Chung, H., Kim, J., Mccann, M.T., Klasky, M.L., Ye, J.C.: Diffusion Posterior Sampling for General Noisy Inverse Problems (May 2024). https://doi.org/10.48550/arXiv.2209.14687, http://arxiv.org/abs/2209.14687, arXiv:2209.14687 [stat]
2. Chung, H., Ryu, D., McCann, M.T., Klasky, M.L., Ye, J.C.: Solving 3D Inverse Problems using Pre-trained 2D Diffusion Models. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 22542–22551 (Jun 2023). https://doi.org/10.1109/CVPR52729.2023.02159, http://arxiv.org/abs/2211.10655, arXiv:2211.10655 [cs]
3. Chung, H., Ye, J.C.: Score-based diffusion models for accelerated MRI (Jul 2022). https://doi.org/10.48550/arXiv.2110.05243, http://arxiv.org/abs/2110.05243, arXiv:2110.05243 [eess]
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (Jun 2009). https://doi.org/10.1109/CVPR.2009.5206848, https://ieeexplore.ieee.org/document/5206848
5. Dhariwal, P., Nichol, A.: Diffusion Models Beat GANs on Image Synthesis (Jun 2021). https://doi.org/10.48550/arXiv.2105.05233, http://arxiv.org/abs/2105.05233, arXiv:2105.05233 [cs]
6. Dou, Z., Song, Y.: Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In: ICLR (2024)
7. Graikos, A., Malkin, N., Jojic, N., Samaras, D.: Diffusion models as plug-and-play priors (Jan 2023). https://doi.org/10.48550/arXiv.2206.09012, http://arxiv.org/abs/2206.09012, arXiv:2206.09012 [cs]
8. Ho, J., Jain, A., Abbeel, P.: Denoising Diffusion Probabilistic Models (Dec 2020). https://doi.org/10.48550/arXiv.2006.11239, http://arxiv.org/abs/2006.11239, arXiv:2006.11239 [cs]
9. Iashchenko, A., Andreev, P., Shchekotov, I., Babaev, N., Vetrov, D.: UnDiff: Unsupervised Voice Restoration with Unconditional Diffusion Model. In: INTERSPEECH 2023. pp. 4294–4298. ISCA (Aug 2023). https://doi.org/10.21437/Interspeech.2023-367, https://www.isca-archive.org/interspeech_2023/iashchenko23_interspeech.html
10. Kadkhodaie, Z., Simoncelli, E.P.: Solving Linear Inverse Problems Using the Prior Implicit in a Denoiser (May 2021). https://doi.org/10.48550/arXiv.2007.13640, http://arxiv.org/abs/2007.13640, arXiv:2007.13640 [cs]
11. Kadkhodaie, Z., Simoncelli, E.P.: Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. In: NeurIPS. vol. 34, pp. 13242–13254 (2021)
12. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the Design Space of Diffusion-Based Generative Models (Oct 2022). https://doi.org/10.48550/arXiv.2206.00364, http://arxiv.org/abs/2206.00364, arXiv:2206.00364 [cs]
13. Karras, T., Laine, S., Aila, T.: A Style-Based Generator Architecture for Generative Adversarial Networks (Apr 2019). https://doi.org/10.48550/arXiv.1812.04948, http://arxiv.org/abs/1812.04948, arXiv:1812.04948 [cs]
14. Kawar, B., Elad, M., Ermon, S., Song, J.: Denoising Diffusion Restoration Models (Oct 2022). https://doi.org/10.48550/arXiv.2201.11793, http://arxiv.org/abs/2201.11793, arXiv:2201.11793 [eess]
15. Kim, H., Kim, S., Yoon, S.: Guided-tts: A diffusion model for text-to-speech via classifier guidance. In: ICML. vol. 162, pp. 11119–11133 (2022)

16. Li, H., Pereira, M.: Solving inverse problems via diffusion optimal control. In: NeurIPS. vol. 37 (2024)
17. Li, J., Wang, C.: Efficient Diffusion Posterior Sampling for Noisy Inverse Problems. SIAM Journal on Imaging Sciences **18**(2), 1468–1492 (Jun 2025). https://doi.org/10.1137/24M1688321, https://epubs.siam.org/doi/10.1137/24M1688321
18. Li, X., Kwon, S.M., Liang, S., Alkhouri, I.R., Ravishankar, S., Qu, Q.: Decoupled Data Consistency with Diffusion Purification for Image Restoration (Jun 2025). https://doi.org/10.48550/arXiv.2403.06054, http://arxiv.org/abs/2403.06054, arXiv:2403.06054 [eess]
19. Mardani, M., Song, J., Kautz, J., Vahdat, A.: A Variational Perspective on Solving Inverse Problems with Diffusion Models (Sep 2023). https://doi.org/10.48550/arXiv.2305.04391, http://arxiv.org/abs/2305.04391, arXiv:2305.04391 [cs]
20. Nie, S., Zhu, F., Du, C., Pang, T., Liu, Q., Zeng, G., Lin, M., Li, C.: Scaling up Masked Diffusion Models on Text (Feb 2025). https://doi.org/10.48550/arXiv.2410.18514, http://arxiv.org/abs/2410.18514, arXiv:2410.18514 [cs]
21. Nie, S., Zhu, F., You, Z., Zhang, X., Ou, J., Hu, J., Zhou, J., Lin, Y., Wen, J.R., Li, C.: Large Language Diffusion Models (Feb 2025). https://doi.org/10.48550/arXiv.2502.09992, http://arxiv.org/abs/2502.09992, arXiv:2502.09992 [cs]
22. Popov, V., Vovk, I., Gogoryan, V., Sadekova, T., Kudinov, M.: Grad-TTS: A Diffusion Probabilistic Model for Text-to-Speech (Aug 2021). https://doi.org/10.48550/arXiv.2105.06337, http://arxiv.org/abs/2105.06337, arXiv:2105.06337 [cs]
23. Romano, Y., Elad, M., Milanfar, P.: The Little Engine that Could: Regularization by Denoising (RED) (Sep 2017). https://doi.org/10.48550/arXiv.1611.02862, http://arxiv.org/abs/1611.02862, arXiv:1611.02862 [cs]
24. Shi, J., Han, K., Wang, Z., Doucet, A., Titsias, M.K.: Simplified and Generalized Masked Diffusion for Discrete Data (Jan 2025). https://doi.org/10.48550/arXiv.2406.04329, http://arxiv.org/abs/2406.04329, arXiv:2406.04329 [cs]
25. Song, J., Meng, C., Ermon, S.: Denoising Diffusion Implicit Models (Oct 2022). https://doi.org/10.48550/arXiv.2010.02502, http://arxiv.org/abs/2010.02502, arXiv:2010.02502 [cs]
26. Song, J., Zhang, Q., Yin, H., Mardani, M., Liu, M.Y., Kautz, J., Chen, Y., Vahdat, A.: Loss-guided diffusion models for plug-and-play controllable generation. In: ICML. vol. 202, pp. 32483–32498 (2023)
27. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. In: NeurIPS. vol. 32 (2019)
28. Song, Y., Shen, L., Xing, L., Ermon, S.: Solving Inverse Problems in Medical Imaging with Score-Based Generative Models (Jun 2022). https://doi.org/10.48550/arXiv.2111.08005, http://arxiv.org/abs/2111.08005, arXiv:2111.08005 [eess]
29. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-Based Generative Modeling through Stochastic Differential Equations (Feb 2021). https://doi.org/10.48550/arXiv.2011.13456, http://arxiv.org/abs/2011.13456, arXiv:2011.13456 [cs]
30. Tran, P., Tran, A., Phung, Q., Hoai, M.: Explore Image Deblurring via Blur Kernel Space (Apr 2021). https://doi.org/10.48550/arXiv.2104.00317, http://arxiv.org/abs/2104.00317, arXiv:2104.00317 [cs]
31. Venkatakrishnan, S.V., Bouman, C.A., Wohlberg, B.: Plug-and-Play priors for model based reconstruction. In: 2013 IEEE Global Conference on Signal and Information Processing. pp. 945–948 (Dec 2013). https://doi.org/10.1109/GlobalSIP.2013.6737048, https://ieeexplore.ieee.org/document/6737048/

32. Vincent, P.: A Connection Between Score Matching and Denoising Autoencoders. Neural Computation **23**(7), 1661–1674 (Jul 2011). https://doi.org/10.1162/NECO_a_00142, https://direct.mit.edu/neco/article/23/7/1661-1674/7677
33. Wang, H., Zhang, X., Li, T., Wan, Y., Chen, T., Sun, J.: DMPlug: A plug-in method for solving inverse problems with diffusion models. In: NeurIPS. vol. 37 (2024)
34. Wang, Y., Yu, J., Zhang, J.: Zero-Shot Image Restoration Using Denoising Diffusion Null-Space Model (Dec 2022). https://doi.org/10.48550/arXiv.2212.00490, http://arxiv.org/abs/2212.00490, arXiv:2212.00490 [cs]
35. Xu, Y., Deng, M., Cheng, X., Tian, Y., Liu, Z., Jaakkola, T.: Restart sampling for improving generative processes. In: NeurIPS. vol. 36 (2023)
36. Zhang, B., Chu, W., Berner, J., Meng, C., Anandkumar, A., Song, Y.: Improving Diffusion Inverse Problem Solving with Decoupled Noise Annealing. In: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 20895–20905 (Jun 2025). https://doi.org/10.1109/CVPR52734.2025.01946, http://arxiv.org/abs/2407.01521, arXiv:2407.01521 [cs]
37. Zhu, Y., Zhang, K., Liang, J., Cao, J., Wen, B., Timofte, R., Gool, L.V.: Denoising Diffusion Models for Plug-and-Play Image Restoration (May 2023). https://doi.org/10.48550/arXiv.2305.08995, http://arxiv.org/abs/2305.08995, arXiv:2305.08995 [cs]