INFERENCE-TIME SEARCH USING SIDE INFORMATION FOR DIFFUSION-BASED IMAGE RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models have emerged as powerful priors for solving inverse problems. However, existing approaches typically overlook side information that could significantly improve reconstruction quality, especially in severely ill-posed settings. In this work, we propose a novel inference-time search algorithm that guides the sampling process using the side information in a manner that balances exploration and exploitation. This enables more accurate and reliable reconstructions, providing an alternative to the gradient-based guidance that is prone to reward-hacking artifacts. Our approach can be seamlessly integrated into a wide range of existing diffusion-based image reconstruction pipelines. Through extensive experiments on a number of inverse problems, such as box inpainting, super-resolution, and various deblurring tasks including motion, Gaussian, nonlinear, and blind deblurring, we show that our approach consistently improves the qualitative and quantitative performance of diffusion-based image reconstruction algorithms. We also show the superior performance of our approach with respect to other baselines, including reward gradient-based guidance algorithms.

1 INTRODUCTION

Diffusion models (Ho et al., 2020; Song et al., 2021b) have demonstrated remarkable success in generative tasks across various fields like text-to-image synthesis (Rombach et al., 2022), protein sequence (Wu et al., 2024), video (Ho et al., 2022), audio (Kong et al., 2021), and language modeling (Austin et al., 2021; Sahoo et al., 2024). Aside from generation, these models have also shown great promise in *solving inverse problems*, where the goal is to reconstruct an image from partial or noisy observations (Chung et al., 2023b; Song et al., 2023a; Rout et al., 2023; Song et al., 2024; He et al., 2024; Ye et al., 2024; Zhang et al., 2024). Inverse problems differ fundamentally from standard generative tasks (e.g., text-to-image synthesis, personalized editing, style transfer): whereas those tasks are often judged subjectively, inverse problems have a precise objective, to recover a specific ground-truth signal from incomplete measurements. Consequently, fidelity to the ground truth and rigorous quantitative evaluation are critical.

When the observation is heavily degraded, the inverse problem becomes highly ill-posed as many distinct signals can explain the data almost equally well. In this regime, unconstrained posterior sampling rarely recovers the ground truth; it tends instead to produce diverse yet semantically inconsistent reconstructions. A practical solution is to incorporate **side information**, auxiliary measurements correlated with the target signal, to constrain the solution space and steer the reconstruction toward faithful outcomes. This idea is well established in the classical signal processing literature, where certain structural or encoded properties are used to guide the iterative algorithms that solve the inverse problem (Jones, 2009; Chun et al., 2012; Oymak et al., 2013; Ehrhardt et al., 2014; Mota et al., 2017; Hyder et al., 2019). In medical imaging, leveraging complementary measurements or modalities, such as multiple MRI contrasts, multimodal microscopy, or RGB guidance for NIR imaging, has been shown to substantially improve quality (Atalık et al., 2025; Tsiligianni & Deligiannis, 2019).

While the existing works on diffusion-based solvers have made significant progress on measurement-only inverse problems, they largely sidestep the harder and increasingly common setting where we must also exploit side information (e.g., a reference photograph of the same person, a text description, or features from another modality). A key obstacle is the challenge of learning the conditional distribution $p_{X|Y,S}$, where X denotes the target image, Y denotes the noisy measurement, and S denotes the side information. While some recent works (Kim et al., 2025a; Chung et al., 2025) address the limited setting of textual side information, these approaches typically train a diffusion

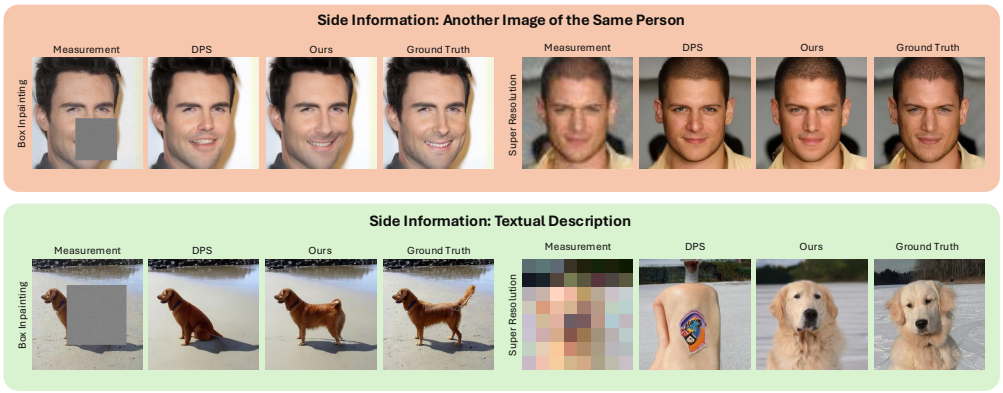


Figure 1: Illustration of the performance of our inference-time search algorithm for using side information in solving inverse problems, compared with the DPS algorithm (Chung et al., 2023b).

model to take a specific side-info modality as input; this demands large paired datasets and expensive training, ties the solver to a single conditioning format, and is impractical when the test-time side information differs from what the model was trained on. This motivates us to address the following question:

How can we leverage a pre-trained (unconditional) diffusion prior to solve inverse problems with side information at inference time, without any retraining, so that the method is modality-agnostic and can use text, images, or features depending on the end-use applications? We provide constructive solutions to these questions in our work. Our main contributions are the following.

- **Modeling:** We introduce a general modeling approach that incorporates arbitrary side information via an auxiliary reward, characterizing $p_{X|S}$ as a reward-tilted version of the pre-trained diffusion prior. This abstraction cleanly decouples the measurement model from the side information, is modality-agnostic (text, image, features), and requires no retraining. We use this modeling with tractable approximations and appropriate error bounds for computing the conditional score functions that are needed for sampling from the pre-trained diffusion models.
- **Algorithm:** Motivated by recent successes of inference-time search in LLMs (Snell et al., 2025; Setlur et al., 2025; Liu et al., 2025), we propose a compute-aware, training-free inference-time search framework that can leverage the side information to solve inverse problems. We instantiate this framework by proposing two specific search algorithms: (i) Greedy Search (GS), a strategy that resamples greedily at each step, and (ii) Recursive Fork-Join Search (RFJS), which balances exploration and exploitation through a group-based sampling at each step. The framework operates as a plug-in on top of any standard inverse-problem solvers and supports black-box, non-differentiable rewards. To our knowledge, our work is the first to propose inference-time search with side information for diffusion-based inverse problems.
- **Experiments:** We provide extensive experimental evaluations of our proposed approach across linear and nonlinear problems (e.g., box inpainting, super-resolution, motion/Gaussian/nonlinear deblurring) and side-information types (images and text), and demonstrate that our approach outperforms multiple relevant baseline algorithms.

2 Related Work

Inverse problems with diffusion priors: Diffusion models (Dhariwal & Nichol, 2021; Ho et al., 2020; Song & Ermon, 2019; Sohl-Dickstein et al., 2015; Song & Ermon, 2020; Song et al., 2021a) are powerful generative models that sample from data distributions by iteratively denoising random noise. Several works adapt diffusion priors to inverse problems via likelihood score approximations. Diffusion Posterior Sampling (DPS) (Chung et al., 2023b) is a foundational method for solving inverse problems in a principled way. Its key idea is to approximate the expected conditional likelihood by evaluating the likelihood at the conditional mean, effectively pushing the expectation through the nonlinear function (Sec. 3.1). ΠGDM (Song et al., 2023a) solves linear inverse problems using a better

approximation than DPS; MPGD (He et al., 2024) avoids this cost by enforcing data consistency in image space; MCG (Chung et al., 2022) constrains reconstructions via manifold projections; DDRM (Kawar et al., 2022) operates in spectral space; and DAPS (Zhang et al., 2024) decouples diffusion steps. Latent diffusion priors are also used: PSLD (Rout et al., 2023) adds consistency terms, ReSample (Song et al., 2024) solves per-step optimization problems, and Chung et al. (2024) tunes prompts for efficiency. These methods, however, do not leverage side information.

Inverse problems with side information: Many works in signal processing (Mota et al., 2017; Oymak et al., 2013; Jones, 2009; Chun et al., 2012; Ehrhardt et al., 2014; Hyder et al., 2019) integrate structural correlations from auxiliary signals, often via designing appropriate optimization algorithms. In MRI, LeSITA (Tsiligianni & Deligiannis, 2019) learns coupled sparse representations, and TGVN (Atalık et al., 2025) constrains ambiguous subspaces with additional contrasts using learned unrolled networks. Diffusion-based approaches include training with joint priors across modalities (Levac et al., 2023; Efimov et al., 2025), metadata conditioning (Chung et al., 2025), and text-guided regularization (Kim et al., 2025a). Most approaches, however, are training-based or bound to one modality of side information associated with the trained conditional diffusion model.

Reward-gradient guidance: LGD (Song et al., 2023b) refines DPS via Monte Carlo estimates, while UGD (Bansal et al., 2024), FreeDoM (Yu et al., 2023), and RB-Modulation (Rout et al., 2025) propose to guide the diffusion with a gradient of the reward function. In addition to being gradient-based approaches, they are typically used for semantic generation tasks rather than inverse problems.

SMC methods: Sequential Monte Carlo approaches (Cardoso et al., 2024; Dou & Song, 2024; Wu et al., 2023) generate and resample particles under tilted distributions, offering gradient-free alternatives but limited performance at small N. DAS (Kim et al., 2025b) combines resampling with gradients for text-to-image tasks. These methods rely only on the measurement to guide the unconditional sampler and do not exploit side information.

Inference-time search: Reward-guided inference-time search has advanced LLM reasoning using Process Advantage Verifiers (PAVs) (Setlur et al., 2025), compute-optimal scheduling (Snell et al., 2025), and reward-guided small models (Liu et al., 2025). Some recent works (Singhal et al., 2025; Li et al., 2025) apply reward-based search in diffusion for text-to-image/protein generation, but do not consider side information or inverse problems.

3 Preliminaries and Problem Formulation

3.1 Preliminaries

Diffusion models: Diffusion models (Ho et al., 2020; Song et al., 2021b) are powerful generative models that enable sampling from an (unknown) distribution through an iterative process. Diffusion models comprise a forward diffusion process and a reverse denoising process. During the forward process, a clean sample from the distribution p_{data} is progressively corrupted by the addition of Gaussian noise at each timestep, transforming the data distribution into pure noise. Conversely, the reverse process trains a denoising neural network to iteratively remove this introduced noise, enabling the reconstruction of samples from the initial data distribution. The forward process is represented by the stochastic differential equation (SDE), $\mathrm{d}\mathbf{x}_t = f(\mathbf{x}_t, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}_t,\ \forall t \in [0, T]$, where x_0 is sampled from p_{data} and $\mathbf{w}_t$ is a Wiener process. Common choices for f, g are $f(\mathbf{x}_t, t) = -(\beta(t)/2)\mathbf{x}_t$ and $g(t) = \sqrt{\beta(t)}$ for some non-negative monotonic increasing function $\beta(\cdot)$ over $[0, T]$. The corresponding reverse process of this SDE is described by (Anderson, 1982; Song et al., 2021b) $\mathrm{d}\mathbf{x}_t = \left(f(\mathbf{x}_t, t) - g^2(t)\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\right)\mathrm{d}t + \mathrm{d}\mathbf{w}_t,\ \forall t \in [T, 0]$, where p_t denotes the marginal probability distribution of $\mathbf{x}_t$, $\mathbf{x}_T$ is sampled according to a standard Gaussian distribution, and $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ represents the *score function*. Since the marginal distribution p_t is unknown, the score function is approximated by a neural network $\mathcal{D}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ via the minimization of a score-matching objective. In practical implementations, the SDE is discretized into T steps, and we define $\alpha_t \triangleq \prod_{s=1}^{t}(1 - \beta_s)$.

Solving inverse problems using diffusion models: An inverse problem consists of recovering an unknown signal $\mathbf{x}_0$ from noisy, partial observations $\mathbf{y} = \mathbf{A}(\mathbf{x}_0) + \sigma_y \mathbf{z}$, where $\mathbf{A}$ is the measurement model, σ_y is the observation noise level, and $\mathbf{z}$ is typically a Gaussian noise. Often, $\mathbf{A}$ is non-injective, i.e., multiple signals $\mathbf{x}_0$ can produce the same measurement $\mathbf{y}$. A standard approach for estimating $\mathbf{x}_0$ is via the Bayesian framework, assuming a prior distribution p_0 over the signal $\mathbf{x}_0$, and sampling from the posterior distribution $\mathbf{x}_0 \sim p_{0|Y}(\cdot \mid \mathbf{y})$. Though $p_{0|Y}(\cdot \mid \mathbf{y})$ is not

known, this sampling can be achieved by running the backward SDE with replacing the original score function with the conditional score function $\nabla_{\mathbf{x}_t} \log p_{t|Y}(\mathbf{x}_t \mid \mathbf{y})$. Using Bayes' theorem, $\nabla_{\mathbf{x}_t} \log p_{t|Y}(\mathbf{x}_t \mid \mathbf{y}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p_{Y|t}(\mathbf{y} \mid \mathbf{x}_t)$. While the score function network $\mathcal{D}_{\boldsymbol{\theta}}$ of the pre-trained diffusion model can be used to approximate the first term, approximating the second term is significantly more challenging, and numerous approaches (Daras et al., 2024) have been proposed to tackle this challenge. In particular, Diffusion Posterior Sampling (DPS) (Chung et al., 2023b) proposes a simple approach to approximate $p_{Y|t}$ as $p_{Y|t}(\mathbf{y} \mid \mathbf{x}_t) = \mathbb{E}_{\mathbf{x}_0 \sim p_{0|t}(\cdot|\mathbf{x}_t)}[p_{Y|0}(\mathbf{y} \mid \mathbf{x}_0)] \approx p_{Y|0}(\mathbf{y} \mid \mathbb{E}_{\mathbf{x}_0 \sim p_{0|t}(\cdot|\mathbf{x}_t)}[\mathbf{x}_0])$, by pushing the expectation inside the nonlinear $p_{Y|0}(\mathbf{y} \mid \cdot)$. The remaining challenge is to compute the conditional mean $\mathbb{E}_{\mathbf{x}_0 \sim p_{0|t}(\cdot|\mathbf{x}_t)}[\mathbf{x}_0] \triangleq \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)$, which is typically tackled by using Tweedie's formula (Efron, 2011), leveraging the fact that $\mathbf{x}_t$ given $\mathbf{x}_0$ is Gaussian. This results in the estimate

$$\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t) = (1/\sqrt{\alpha_t})(\mathbf{x}_t + (1-\alpha_t)\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)) \approx (1/\sqrt{\alpha_t})(\mathbf{x}_t + (1-\alpha_t)\mathcal{D}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)). \quad (1)$$

3.2 Problem formulation: solving inverse problems with side information

In many applications, the observation $\mathbf{y}$ alone is insufficient to identify the latent signal $\mathbf{x}_0$; auxiliary side information $\mathbf{s}$ (e.g., a reference image, identity/text embedding, or physics-derived features) can dramatically reduce ambiguity. Formally, when side information $\mathbf{s}$ is available, *the goal is to sample from the target conditional distribution* $p_{0|Y,S}(\cdot \mid \mathbf{y}, \mathbf{s})$. A seemingly direct route is to *train a conditional diffusion model* that accepts $\mathbf{s}$ as input, learn the conditional score function $\nabla_{\mathbf{x}_t} \log p_{t|S}(\mathbf{x}_t \mid \mathbf{s})$, and then approximate the full conditional score $\nabla_{\mathbf{x}_t} \log p_{t|Y,S}(\mathbf{x}_t \mid \mathbf{y}, \mathbf{s}) = \nabla_{\mathbf{x}_t} \log p_{t|S}(\mathbf{x}_t \mid \mathbf{s}) + \nabla_{\mathbf{x}_t} \log p_{Y|t,S}(\mathbf{y} \mid \mathbf{x}_t, \mathbf{s})$ through a DPS-style method for the second term, to run the backward SDE. However, this training-based approach is often impractical: it demands large paired datasets $(\mathbf{x}_0, \mathbf{s})$, which are expensive or impossible to curate; it locks the solver to the training modality of $\mathbf{s}$ (a text-conditioned prior cannot natively exploit an image or spectral feature at test time); and general multi-modal conditioning requires prohibitive data and compute. These constraints motivate a training-free alternative that reuses strong unconditional diffusion priors and uses $\mathbf{s}$ only at inference, preserving modality-agnosticism and avoiding costly data collection.

Designing such a training-free method is technically challenging. First, DPS-style derivations rely on tractable likelihoods (e.g., Gaussian $p_{Y|0}$), whereas realistic $p_{S|0}$ are often non-Gaussian implicitly, complicating conditional-score construction. Second, even for measurement-only guidance, computing the conditional score used in the DPS-style algorithms requires back-propagating through the denoiser at every step. Naively extending to side information forces second-order/Hessian terms through the diffusion network. Third, purely gradient-guided diffusion is brittle: it struggles with non-differentiable or black-box rewards, amplifies early-step errors, and can drift off the data manifold. Inference-time search approaches, which have shown remarkable performance improvement in LLMs (Setlur et al., 2025; Liu et al., 2025; Snell et al., 2025) and text-conditioned diffusion models (Singhal et al., 2025; Kim et al., 2025b), but have not yet been used for solving the inverse problems, offer a promising path to overcome these challenges. In this context, we address the following questions:

(i) *Modeling:* How can we realize $p_{0|Y,S}$ at inference time, without any retraining, by constructing a surrogate objective that is valid across diverse side-information modalities? (ii) *Algorithm*: How can we design a plug-and-play inference-time search module that is modality-agnostic, compute-aware, and capable of making global corrections (beyond local gradient steps)?

4 Modeling and Algorithm

4.1 Modeling side information using reward function

Given a side-information signal $\mathbf{s}$ corresponding to an unknown $\mathbf{x}_0$, and two candiate reconstructions, $\mathbf{x}_0^1$ and $\mathbf{x}_0^2$, a principled way to decide which reconstruction is more truthful is to compare the (unknown) conditional probabilities $p_{0|S}(\mathbf{x}_0^1 \mid \mathbf{s})$ and $p_{0|S}(\mathbf{x}_0^2 \mid \mathbf{s})$. Directly estimating $p_{0|S}$ is intractable in our setting: it is typically non-Gaussian, multi-modal, and depends on the data domain and modality of $\mathbf{s}$. We therefore introduce a reward function $r : \mathbb{R}^d \times \mathcal{S} \to \mathbb{R}$ that orders reconstructions given $\mathbf{s}$: if $r(\mathbf{x}_0^1, \mathbf{s}) > r(\mathbf{x}_0^2, \mathbf{s})$, then $\mathbf{x}_0^1$ is deemed more compatible with $\mathbf{x}_0$ than $\mathbf{x}_0^2$. This abstraction aligns with many real-world applications (as shown in our experiments): when $\mathbf{s}$ is a text description of the target image $\mathbf{x}_0$, we can use a pre-trained text-image model to score text-image alignment. When $\mathbf{s}$ is a reference image of the same entity (e.g., the same person under different poses/lighting), we can use a pre-trained network to score image-image similarity. Such pre-trained

rewards are typically available across datasets, and monotone with respect to the intuitive notion of *agreement* with $\mathbf{x}_0$. In this sense, they serve as practically justified surrogates for comparing $p_{0|S}$ without requiring an explicit conditional density model.

Our key modeling choice is to use $r(\cdot, \mathbf{s})$ to implicitly characterize $p_{0|S}(\cdot \mid \mathbf{s})$ by tilting the unconditional prior p_0 toward higher-reward regions. Our approach is inspired by the alignment framework used in LLMs (Ouyang et al., 2022; Rafailov et al., 2023), where the goal is to generate a sample $\mathbf{x}$ that maximizes some reward $r(\mathbf{x})$, while ensuring that the sampling distribution does not deviate too much from the pre-trained distribution p_0. This is typically formalized as a KL-regularized reward maximization problem, $\max_{p \in \mathcal{P}} \left(\mathbb{E}_{\mathbf{x} \sim p}[r(\mathbf{x})] - \tau D_{\mathrm{KL}}(p \| p_0) \right)$, where $\tau > 0$ offers the trade-off between the deviation from the prior and reward maximization. This optimization problem admits a closed-form solution, $p^*(\mathbf{x}) \propto p_0(\mathbf{x}) \exp(r(\mathbf{x})/\tau)$ (Rafailov et al., 2023). Based on this intuition, we make the following *modeling assumption*: the conditional distribution $p_{0|S}$ is approximated as,

$$p_{0|S}(\mathbf{x}_0 \mid \mathbf{s}) \propto p_0(\mathbf{x}_0) \exp\left(\tfrac{r(\mathbf{x}_0; \mathbf{s})}{\tau} \right), \tag{2}$$

This assumption: (i) preserves the powerful unconditional diffusion prior p_0, (ii) injects modality-agnostic side information via a reward, and (iii) produces a tractable objective that we can combine with the measurement model to target $p_{0|Y,S}$ at inference time using a pre-trained diffusion model. We do not claim optimality of Eq. (2); rather, we show it leads to a practical, training-free algorithm that consistently improves reconstructions over strong baselines while keeping compute comparable.

We now leverage Eq. (2) to compute the conditional posteriors for the reverse diffusion.

Proposition 1. *Let $p_{t|t+1,Y,S}$ denote the conditional posterior distribution for the reverse diffusion process. Then using (2) we have*

$$p_{t|t+1,Y,S}(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{y}, \mathbf{s}) \propto p_{t|t+1,Y}(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{y}) \exp(V_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y})), \tag{3}$$

$$p_{t|Y,S}(\mathbf{x}_t \mid \mathbf{y}, \mathbf{s}) \propto p_{t|Y}(\mathbf{x}_t \mid \mathbf{y}) \exp(V_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y})), \tag{4}$$

where $V_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y}) \triangleq \log \mathbb{E}_{\mathbf{x}_0 \sim p_{0|t,Y}(\cdot|\mathbf{x}_t, \mathbf{y})}[\exp(r(\mathbf{x}_0; \mathbf{s})/\tau)]$.

The proof is provided in Appendix A.1. Using (4), we can get the conditional score function as,

$$\nabla_{\mathbf{x}_t} \log p_{t|Y,S}(\mathbf{x}_t \mid \mathbf{y}, \mathbf{s}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p_{Y|t}(\mathbf{y} \mid \mathbf{x}_t) + \nabla_{\mathbf{x}_t} V_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y}). \tag{5}$$

The computation of V_t^τ is not straightforward. So, we use a DPS-style approximation as $V_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y}) = \log \mathbb{E}_{\mathbf{x}_0 \sim p_{0|t,Y}(\cdot|\mathbf{x}_t, \mathbf{y})}[\exp(r(\mathbf{x}_0; \mathbf{s})/\tau)] \approx r(\mathbb{E}_{\mathbf{x}_0 \sim p_{0|t,Y}(\cdot|\mathbf{x}_t, \mathbf{y})}[\mathbf{x}_0]; \mathbf{s})/\tau = r(\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t, \mathbf{y}); \mathbf{s})/\tau$. Using some approximation and the fact that $p_{Y|0}$ is Gaussian, we can get

$$\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t, \mathbf{y}) \approx \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t) - (1 - \alpha_t)(\sqrt{\alpha_t}))\eta \nabla_{\mathbf{x}_t} \|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)\|_2^2, \tag{6}$$

$$V_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y}) \approx \hat{V}_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y}) \triangleq r\left(\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t, \mathbf{y}); \mathbf{s}\right)/\tau. \tag{7}$$

In Appendix A.2, we have provided the details of the steps leading to Eq. (6)-Eq. (7).

We characterize the error in approximating the value function, $|V_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y}) - \hat{V}_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y})|$, in Proposition 3, which is deferred to Appendix A.2.

We can now get $\nabla_{\mathbf{x}_t} \log p_{t|Y,S}(\mathbf{x}_t \mid \mathbf{y}, \mathbf{s})$ given in Eq. (5) by replacing $\nabla_{\mathbf{x}_t} V_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y})$ with $\nabla_{\mathbf{x}_t} \hat{V}_t^\tau(\mathbf{x}_t; \mathbf{y}, \mathbf{s})$. However, running a backward diffusion using $\nabla_{\mathbf{x}_t} \hat{V}_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y})$ is computationally infeasible because it involves computing second-order derivatives through the denoiser network. This issue, however, can be circumvented by making a further approximation, by setting $\eta = 0$ in Eq. (6) to get $\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t, \mathbf{y}) \approx \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)$, which leads to the approximation $\nabla_{\mathbf{x}_t} V_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y}) \approx \nabla_{\mathbf{x}_t} r(\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t); \mathbf{s})$. We show that approximation error remains small when t is small in Appendix A.2 even when $\eta = 0$. This approach then reduces to the **reward gradient guidance (RGG)** approach used for the inference-time alignment of diffusion models (Bansal et al., 2024; Kim et al., 2025b; Yu et al., 2023; He et al., 2024), with the critical difference being that the guidance is from both $\mathbf{s}$ and $\mathbf{y}$.

The RGG approach, however, is limited only to differentiable rewards, and even when they are differentiable, calculating a gradient through the denoiser network at each step of the backward diffusion is computationally intensive and can be ill-suited for many end-use edge-device applications. Moreover, the hyperparameter that determines the weight of the reward gradient guidance is highly sensitive and is difficult to tune, leading to limited performance improvements and undesirable artifacts in the reconstructed images. We later illustrate these issues in Appendix B.4. This motivates us to pursue a gradient-free approach for leveraging the side information for inverse problems.

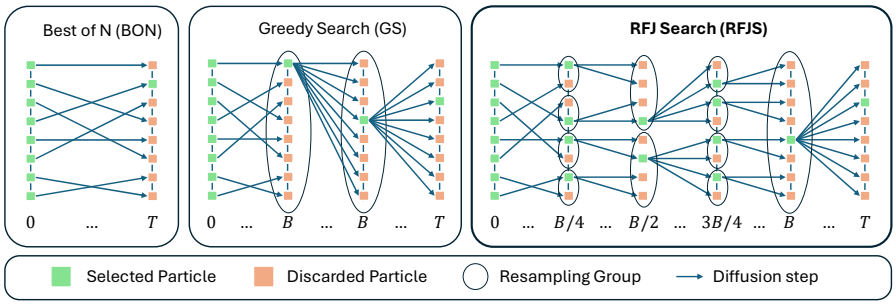


Figure 2: Illustration of the group size resampling strategies of different search algorithms.

4.2 Inference-time Search Algorithms for Inverse Problems

Inference-time search methods have recently gained traction as a means of improving the quality of output from LLMs (Snell et al., 2025; Setlur et al., 2025; Liu et al., 2025). The key objective of a search algorithm is to solve a multi-step decision-making problem with balanced exploration and exploitation. While Monte Carlo Tree Search (MCTS) (Kocsis & Szepesvári, 2006) was successful in large-scale reinforcement learning systems like AlphaGo (Silver et al., 2016), they are infeasible for diffusion models: estimating the expected reward of a noisy state $\mathbf{x}_t$ would require repeated rollouts of the reverse process. Training a value function to guide the search rewards is another alternative, but this demands additional training customized to each modality and type of side information. These limitations motivate inference-time methods that are both training-free and computationally tractable.

Particle-based procedures offer one such approach, using the distribution given by Eq. (3), where the value function is replaced by the approximation in Eq. (7). At a given step, suppose we have N samples $\mathbf{x}_{t+1}[1], \ldots, \mathbf{x}_{t+1}[N] \sim p_{t+1|Y,S}$. One way to generate samples from $p_{t|Y,S}$ is to (i) propose candidates $\tilde{\mathbf{x}}_t[i] \sim p_{t|t+1,Y}(\cdot \mid \mathbf{x}_{t+1}[i], \mathbf{y})$, (ii) compute rewards $r[i] = r(\hat{\mathbf{x}}_{0|t,Y}(\tilde{\mathbf{x}}_t[i], \mathbf{y}); \mathbf{s})$ (approximate value) (iii) assign weights $w[i] \propto \exp(r[i]/\tau)$ and resample indices with replacement $I[i] \sim \mathrm{Cat}(w[1:N])$, and (iv) retain $\mathbf{x}_t[I[i]]$ for the next step. In theory, such particle methods converge to the target distribution as $N \to \infty$ and with exact tilting (Wu et al., 2023; Dou & Song, 2024). In practice, however, finite N and approximate tilting has some issues: frequent resampling favors exploitation but risks reward over-optimization, while no resampling preserves data consistency but requires prohibitively many particles to harness and optimize the reward.

To address this trade-off, we modify the step (iii) by introducing **grouped resampling** at each time step t, where particles are resampled within groups of size g_t. Formally, let the index set of the i-th group be $\mathcal{G}_i = \{(i-1)g_t + 1, \ldots, ig_t\}$ for $i = 1, \ldots, N/g_t$. For each group, we draw g_t indices with replacement according to the weights within that groups, i.e., $I'[j] \sim \mathrm{Cat}(w[\mathcal{G}_i]) \in [1 : g_t]$. The absolute indices are then obtained by shifting I' as $I[(i-1)g_t + j : ig_t + j] = (i-1)g_t + I'[j]$ for $j = 1, \ldots, g_t$. Based on the choice of g_t, we introduce two specific search strategies.

Greedy Search (GS): Here, we use a fixed resampling period B and select $g_t = N$ whenever $t \bmod B = 0$, and $g_t = 1$ otherwise. Greedy Search reduces to the **Best-of-N** (BON) strategy when $B \geq T$, since in that case $g_t = 1$ for all t. Smaller values of B emphasize short-term reward exploitation, while larger values promote long-term consistency and exploration. An illustration of Greedy Search, with resampling interval B, is provided in Figure 2, where the particles evolve independently between resampling events and only interact at steps that are multiples of B.

Recursive Fork-Join Search (RFJS): Greedy search considers the largest resampling group size (N) at fixed time periods of B, and greedily selects one particle from this group, which leads to an exploitation-style approach in search. Selecting the smallest group size $(g_t = 1)$ leads to a pure exploration-style search of BON. Ideally, one should combine the benefits of resampling with multiple group sizes at multiple time steps to get a balanced exploration and exploitation.

To this end, we propose a *recursive grouping and sampling approach*, which we call recursive fork-join search (RFJS), in which the resampling group sizes vary systematically over time. At every B steps, all N particles are resampled together; at every $B/2$ steps, the particles are partitioned into groups of size $N/2$ that are resampled independently; at every $B/4$ steps, groups of size $N/4$ are resampled; and so on. This hierarchical schedule is illustrated in Figure 2. As a concrete example,

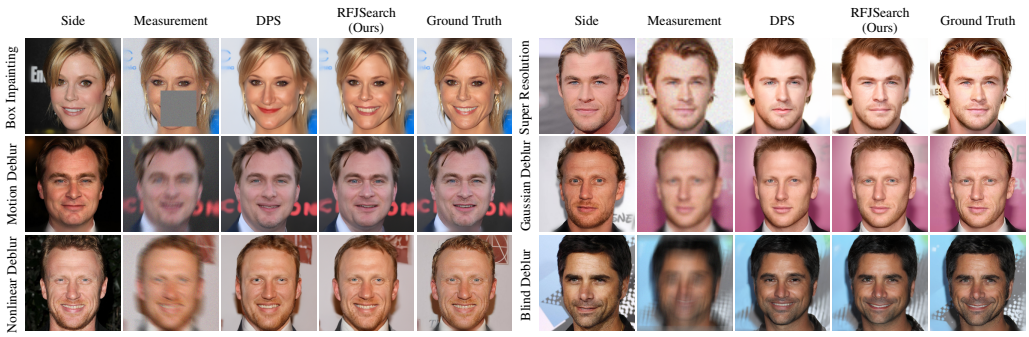


Figure 3: **Image as side information:** Qualitative illustration of the performance of our RFJS algorithm compared to the DPS baseline on linear and nonlinear inverse problems. RFJS is able to capture many details that are missed by the DPS baseline to achieve a superior reconstruction quality.

	Box Inpainting				Super Resolution (×4)				Non-linear Deblur			
Algorithm	FS (↓)	PSNR (↑)	LPIPS (↓)	SSIM (↑)	FS (↓)	PSNR (↑)	LPIPS (↓)	SSIM (↑)	FS (↓)	PSNR (↑)	LPIPS (↓)	SSIM (↑)
RFJS (ours)	**0.308**	**28.29**	**0.136**	**0.855**	**0.380**	**25.26**	**0.225**	0.695	**0.394**	23.89	**0.229**	0.668
GS (ours)	0.349	28.22	0.137	0.855	0.460	25.24	0.225	**0.696**	0.467	**23.92**	0.232	**0.669**
RGG	0.475	27.96	0.138	0.851	0.573	25.13	0.228	0.690	0.654	23.89	0.231	0.665
BON	0.584	28.20	0.137	0.854	0.915	25.14	0.229	0.694	0.881	23.89	0.233	0.667
DPS	0.739	27.93	0.139	0.852	1.042	25.13	0.229	0.693	1.008	23.87	0.232	0.666
	Motion Deblur				Gaussian Deblur				Blind Deblur			
RFJS (ours)	**0.326**	**26.64**	**0.193**	**0.736**	**0.330**	**26.20**	**0.196**	**0.712**	**0.341**	25.04	**0.209**	0.707
GS (ours)	0.392	26.58	0.193	0.735	0.385	26.16	0.198	0.711	0.417	25.04	0.211	0.706
RGG	0.497	26.55	0.193	0.733	0.495	26.15	0.200	0.709	0.473	24.97	0.211	0.701
BON	0.671	26.57	0.194	0.735	0.667	26.18	0.201	0.711	0.642	**25.15**	0.210	**0.708**
DPS	0.815	26.54	0.194	0.734	0.807	26.15	0.200	0.711	0.779	24.98	0.213	0.704

Table 1: **Image as side information:** Quantitative comparison of our GS and RFJS algorithms with baseline algorithms. For each evaluation metric, the best result is shown in **bold**, and the second best is underlined. RFJS and GS achieve superior performance consistently across all tasks and metrics.

consider $N = 8$. In this case, groups of size at least $N/4 = 2$ are resampled every $B/4$ steps, groups of size $N/2 = 4$ are resampled every $B/2$ steps, and all $N = 8$ particles are resampled every B steps. When multiple group sizes are scheduled to be resampled at the same time step t, the larger group size always takes precedence. For example, although $t = B/2$ is also a multiple of $B/4$, the scheme prioritizes the larger group size. Thus, rather than resampling groups of size 2, we resample groups of size 4 at $t = B/2$. Similarly, at $t = B$, the entire set of $N = 8$ particles is resampled jointly. More generally, the group size at time step t is given by $g_t = N/2^{j^*}$, where $j^* = \min\{\, i \geq 0 : \; t \bmod (B/2^i) = 0 \,\}$. The localized resampling (fork) at intermediate group sizes encourages balanced exploration, while the recursive return to larger group sizes (join) encourages exploitation. Naively reducing B in GS does not balance this trade-off well and may lead to an undesirable compromise between exploration and exploitation.

We have summarized this inference-time search framework in Algorithm 1 in the Appendix A.3. Our framework is modular: the resampling rule, whether BON, GS, or RFJS, can be chosen depending on budget and application. Since this requires no retraining and works with arbitrary reward functions, it can be incorporated into any diffusion-based inverse problem solvers with minimal modification.

5 Experiments

5.1 Experimental Setup

We evaluate our inference-time search framework for solving inverse problems with side information by instantiating two specific search algorithms we proposed: **Greedy Search (GS)** and **Recursive Fork Join Search (RFJS)**, both of which are described in the previous section. We consider two types of side information: (i) **image as side information**, where a reference image of the same entity (here, the same person under different poses/lighting) is used as side information, and (ii) **text as side information**, where a text description of the target image is used as a side information.

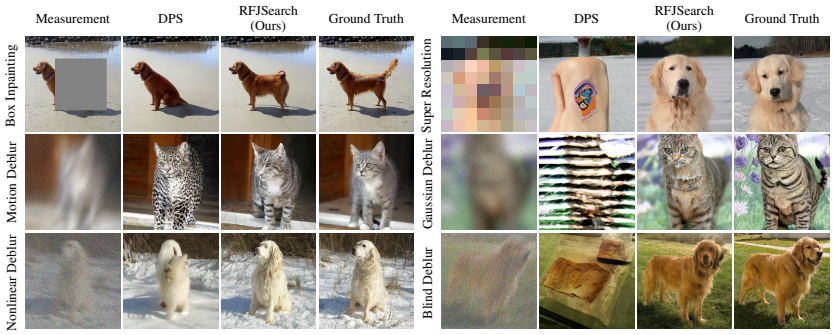


Figure 4: **Text as side information :** Qualitative illustration of the performance of our RFJS algorithm compared to the DPS baseline. For example, the side information provided for the super resolution task is *'golden retriever sitting on a snowy frozen lake, facing forward'*. RFJS is able to capture many details that are missed by the DPS baseline to achieve a superior reconstruction quality.

	Box Inpainting				Super Resolution (×32)				Non-linear Deblur			
Algorithm	CS (↑)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CS (↑)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CS (↑)	PSNR (↑)	SSIM (↑)	LPIPS (↓)
RFJS (ours)	**0.901**	**20.75**	**0.678**	**0.294**	**0.801**	17.13	**0.352**	**0.4926**	0.863	**20.58**	**0.473**	**0.405**
GS (ours)	0.894	19.76	0.676	0.305	0.791	17.20	0.351	0.5094	**0.865**	20.32	0.456	0.405
BON	0.882	19.99	0.672	0.308	0.788	**17.21**	0.350	0.5003	0.855	20.52	0.464	0.406
DPS	0.871	19.86	0.672	0.312	0.731	16.90	0.330	0.5220	0.839	20.55	0.469	0.409
	Motion Deblur				Gaussian Deblur				Blind Deblur			
RFJS (ours)	**0.858**	18.61	0.402	**0.424**	**0.843**	**18.10**	0.358	0.457	**0.851**	18.84	0.412	**0.433**
GS (ours)	0.835	17.83	0.369	0.453	0.835	17.96	0.356	0.457	0.835	**18.93**	**0.414**	0.438
BON	0.848	**19.24**	**0.415**	0.427	0.831	17.99	**0.365**	**0.452**	0.831	18.78	0.410	0.443
DPS	0.794	18.16	0.384	0.458	0.778	16.79	0.329	0.487	0.793	18.82	0.409	0.459

Table 2: **Text as side information:** Quantitative comparison of our GS and RFJS algorithms with baseline algorithms. RFJS and GS achieve better performance across all tasks and metrics.

We demonstrate the plug-and-play nature of our algorithms by considering four different baseline inverse problem solvers: (i) **DPS** (Chung et al., 2023b), (ii) **BlindDPS** (Chung et al., 2023a), (iii) **MPGD** (He et al., 2024), and (iv) **DAPS** (Zhang et al., 2024). Due to page limitation, the evaluation results using DAPS and MPGD are deferred to Appendix B.

Inverse problems: We evaluate our algorithms on six inverse problems, covering both linear and nonlinear problems. The linear problems are: (i) box inpainting, (ii) super resolution, (iii) motion deblurring, and (iv) Gaussian deblurring. The nonlinear problems are: (v) nonlinear deblurring, and (vi) blind deblurring. A detailed description of these inverse problems is given in Appendix C.

Baselines: We compare the performance of GS and RFJS against the following baselines: (i) *Baseline solvers (DPS, BlindDPS, MPGD, DAPS)*, (ii) *Best-of-N (BoN)*, which generates N independent samples and selects the one with the best reward at the end, (iii) *Reward Gradient Guidance (RGG)*, which solves the inverse problem by running the backward diffusion according to Eq. (5), but with the approximation $\nabla_{\mathbf{x}_t} V_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y}) \approx \nabla_{\mathbf{x}_t} r(\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t); \mathbf{s})$. Unless otherwise noted, hyperparameters, including guidance scale, number of diffusion steps, and task-specific settings, match the original baseline implementations. The specific values of hyperparamaters are listed in Appendix C. All the experiments are run on NVIDIA A100 GPUs on an internal compute cluster.

5.2 Main Results

Image as side information: The goal is to reconstruct a human face from a noisy observation when another image of the same identity is available (Fig. 3). Using Celeb-HQ (Na et al., 2022) as an out-of-distribution set and a diffusion model pretrained on FFHQ (Chung et al., 2023b), we sample two random images per identity for target and side information. We compute the reward as follows: first, detect the face using MTCNN (Zhang et al., 2016) and then extract identity features with AdaFace (Kim et al., 2022). Then, we measure the reward as the negative of the FaceSimilarity (FS) loss, computed as the distance between the identity embeddings of the reconstructed and side-information faces, extracted by pretrained AdaFace network. We evaluate on 64 pairs, using $N = 8$ particles and $B = 16$, with a gradient scale 0.5 for RGG. We evaluate with standard metrics, PSNR, SSIM, and LPIPS, but these often fail to measure the identity similarity. Thus, we use FaceSimilarity (FS),

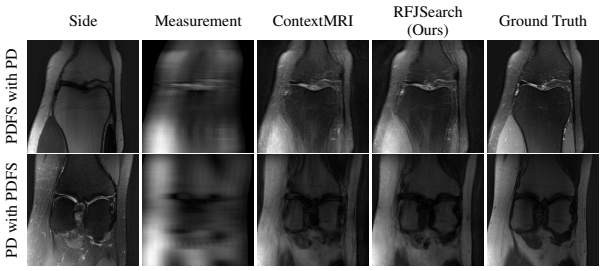


Figure 5: **Contrast Image as Side Information:** Qualitative MRI reconstruction with RFJS vs. ContextMRI. The shapes and line edges are well preserved in our reconstruction.

Algorithm	PSNR (↑)	SSIM (↑)	LPIPS (↓)	NMI (↑)
PDFS with PD				
RFJS	**25.85**	**0.801**	**0.375**	**0.457**
GS	25.33	0.797	0.375	0.455
BON	25.47	0.797	0.376	0.454
ContextMRI	25.39	0.795	0.383	0.451
PD with PDFS				
RFJS	**27.85**	**0.920**	**0.358**	**0.579**
GS	27.80	0.920	0.360	0.579
BON	27.80	0.918	0.366	0.570
ContextMRI	27.46	0.915	0.375	0.563

Figure 6: Quantitative MRI reconstruction results (fastMRI knee, AF=16, ACS=2%).

comparing the reconstruction to the ground truth for a more reliable measure of identity preservation. Table 1 shows that both proposed inference-time search methods, GS and RFJS, outperform baselines, with RFJS achieving the best overall scores indicating a stronger balance between exploration and exploitation. Qualitative results given in Fig 3 show sharper facial details and preserved identity traits, whereas Fig 7 in Appendix B.1 indicates the importance of the FS metric.

Text as side information: The goal is to reconstruct an image from its noisy observation, with a text description of the image available as side information. We use a pre-trained diffusion model trained on the ImageNet data (Dhariwal & Nichol, 2021). We use 25 images from the ImageNet validation set to evaluate the algorithms and generated a short one-sentence textual description for each image using ChatGPT. We use ImageReward (Xu et al., 2023), a pre-trained network that measures text-to-image similarity, as the reward function. We consider some inverse problem tasks that are significantly challenging, including $\times 32$ super resolution, and strong blur with larger kernels. Experiments use $N{=}4$ and $B{=}100$, and we report the standard metrics and CLIPScore (Radford et al., 2021). CLIPScore measures the cosine similarity between CLIP image embeddings of the ground truth and reconstruction, providing a semantically informed metric that reflects both visual and textual alignment. It can be seen in Fig. 4 that the qualitative reconstructions closely match textual descriptions. The quantitative metrics are in Table 2 where both GS and RFJS outperform competing baselines, with RFJS achieving the highest CLIPScore.

MRI with multi-contrast side information: Finally, we test on fastMRI knee dataset (Zbontar et al., 2018) with the ContextMRI model (Chung et al., 2025). We pair PD and PDFS contrasts, reconstructing one from the other under highly accelerated $16\times$ undersampling with 2% ACS. We use normalized mutual information (NMI) as reward, which is robust to contrast changes. Table 6 shows our methods consistently outperform the baseline in all the metrics of interest. Figure 5 highlights sharper edges and more faithful structure.

Additional Experiments: To demonstrate the generality of our framework, we extended our search algorithm beyond DPS to other samplers, including DAPS and MPGD. Qualitative and quantitative results for these experiments are provided in Appendix B, along with additional DPS results for both types of side information. We also conducted several ablations to analyze the role of side information and the scalability of our approach. Appendix B.4 studies the sensitivity of the gradient-guided methods. Appendix B.5 examines the effect of the number of particles: increasing N improves exploration and reward, while runtime grows sublinearly due to parallelization (Appendix B.6). To build intuition, Appendix B.7 provides 2D toy examples illustrating the benefits of side information and the impact of B; Fig. 20 further shows that RFJS is more robust than GS when the reward is non-linear and non-convex. Hyperparameter details are summarized in Appendix C.

6 Conclusion

We proposed a lightweight, modular inference-time search algorithm that integrates side information into diffusion-based image reconstruction, in a principled way. By adaptively guiding the generative process, our method delivers substantial quality gains, especially in ill-posed settings, while requiring only minimal changes to existing pipelines. Extensive experiments across standard reconstruction tasks show consistent improvements in both visual fidelity and quantitative metrics, and our approach surpasses gradient-based alternatives. These results highlight the power of leveraging side information at inference time to make diffusion-based solvers more reliable and accurate.

REFERENCES

Brian D.O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. ISSN 0304-4149. doi: https://doi.org/10.1016/0304-4149(82)90051-5. URL https://www.sciencedirect.com/science/article/pii/0304414982900515.

Arda Atalık, Sumit Chopra, and Daniel Sodickson. A Trust-Guided Approach to MR Image Reconstruction with Side Information. *IEEE Transactions on Medical Imaging*, 2025.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, 2021.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Roni Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal Guidance for Diffusion Models. In *The Twelfth International Conference on Learning Representations*, 2024.

Gabriel Cardoso, Yazid Janati el idrissi, Sylvain Le Corff, and Eric Moulines. Monte Carlo guided Denoising Diffusion models for Bayesian linear inverse problems. In *The Twelfth International Conference on Learning Representations*, 2024.

Se Young Chun, Jeffrey A. Fessler, and Yuni K. Dewaraja. Non-local means methods using ct side information for i-131 spect image reconstruction. In *2012 IEEE Nuclear Science Symposium and Medical Imaging Conference Record (NSS/MIC)*, 2012.

Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving Diffusion Models for Inverse Problems using Manifold Constraints. In *Advances in Neural Information Processing Systems*, 2022.

Hyungjin Chung, Jeongsol Kim, Sehui Kim, and Jong Chul Ye. Parallel Diffusion Models of Operator and Image for Blind Inverse Problems . In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023a.

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

Hyungjin Chung, Jong Chul Ye, Peyman Milanfar, and Mauricio Delbracio. Prompt-tuning latent diffusion models for inverse problems. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Hyungjin Chung, Dohun Lee, Zihui Wu, Byung-Hoon Kim, Katherine L Bouman, and Jong Chul Ye. Contextmri: Enhancing compressed sensing mri through metadata conditioning. *arXiv preprint arXiv:2501.04284*, 2025.

Giannis Daras, Hyungjin Chung, Chieh-Hsin Lai, Yuki Mitsufuji, Jong Chul Ye, Peyman Milanfar, Alexandros G Dimakis, and Mauricio Delbracio. A survey on diffusion models for inverse problems. *arXiv preprint arXiv:2410.00083*, 2024.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, 2021.

Zehao Dou and Yang Song. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In *The Twelfth International Conference on Learning Representations*, 2024.

Timofey Efimov, Harry Dong, Megna Shah, Jeff Simmons, Sean Donegan, and Yuejie Chi. Leveraging multimodal diffusion models to accelerate imaging with side information. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

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

Matthias J Ehrhardt, Kris Thielemans, Luis Pizarro, David Atkinson, Sébastien Ourselin, Brian F Hutton, and Simon R Arridge. Joint reconstruction of pet-mri by exploiting structural similarity. *Inverse Problems*, 2014.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J Zico Kolter, Ruslan Salakhutdinov, and Stefano Ermon. Manifold preserving guided diffusion. In *The Twelfth International Conference on Learning Representations*, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

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

Rakib Hyder, Chinmay Hegde, and M. Salman Asif. Fourier phase retrieval with side information using generative prior. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, 2019.

Lee K. Jones. Local minimax learning of functions with best finite sample estimation error bounds: Applications to ridge and lasso regression, boosting, tree learning, kernel machines, and inverse problems. *IEEE Transactions on Information Theory*, 2009.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*, 2022.

Jeongsol Kim, Geon Yeong Park, Hyungjin Chung, and Jong Chul Ye. Regularization by texts for latent diffusion inverse solvers. In *International Conference on Learning Representations*, 2025a.

Minchul Kim, Anil K Jain, and Xiaoming Liu. Adaface: Quality adaptive margin for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time Alignment of Diffusion Models without Reward Over-optimization. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *Proceedings of the 17th European Conference on Machine Learning*, 2006.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.

Brett Levac, Ajil Jalal, Kannan Ramchandran, and Jonathan I. Tamir. MRI Reconstruction with Side Information using Diffusion Models. *2023 57th Asilomar Conference on Signals, Systems, and Computers*, 2023.

Xiner Li, Masatoshi Uehara, Xingyu Su, Gabriele Scalia, Tommaso Biancalani, Aviv Regev, Sergey Levine, and Shuiwang Ji. Dynamic search for inference-time alignment in diffusion models. *arXiv preprint arXiv:2503.02039*, 2025.

Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling. *arXiv preprint arXiv:2502.06703*, 2025.

João F. C. Mota, Nikos Deligiannis, and Miguel R. D. Rodrigues. Compressed sensing with prior information: Strategies, geometry, and bounds. *IEEE Transactions on Information Theory*, 2017.

Dongbin Na, Sangwoo Ji, and Jong Kim. Unrestricted black-box adversarial attack using gan with limited queries. In *European Conference on Computer Vision*, 2022.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

Samet Oymak, Christos Thrampoulidis, and Babak Hassibi. Simple bounds for noisy linear inverse problems with exact side information. *arXiv preprint arXiv:1312.0641*, 2013.

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*, 2021.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

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. In *Advances in Neural Information Processing Systems*, 2023.

Litu Rout, Yujia Chen, Nataniel Ruiz, Abhishek Kumar, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. RB-modulation: Training-free stylization using reference-based modulation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, 2024.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for LLM reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Sergey Ioffe, Alan Green, Xi Chen, and et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 2016.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

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.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*, 2021a.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-Guided Diffusion Models for Inverse Problems. In *International Conference on Learning Representations*, 2023a.

Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-Guided Diffusion Models for Plug-and-Play Controllable Generation. In *Proceedings of the 40th International Conference on Machine Learning*, 2023b.

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

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems*, 2020.

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

Evaggelia Tsiligianni and Nikos Deligiannis. Deep Coupled-Representation Learning for Sparse Linear Inverse Problems With Side Information. *IEEE Signal Processing Letters*, 2019.

Kevin E Wu, Kevin K Yang, Rianne van den Berg, Sarah Alamdari, James Y Zou, Alex X Lu, and Ava P Amini. Protein structure generation via folding diffusion. *Nature communications*, 2024.

Luhuan Wu, Brian L. Trippe, Christian A Naesseth, John Patrick Cunningham, and David Blei. Practical and asymptotically exact conditional sampling in diffusion models. In *Advances in Neural Information Processing Systems*, 2023.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and Evaluating Human Preferences for Text-to-Image Generation. In *Advances in Neural Information Processing Systems*, 2023.

Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Zou, and Stefano Ermon. TFG: Unified Training-Free Guidance for Diffusion Models. In *Advances in Neural Information Processing Systems*, 2024.

Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. FreeDoM: Training-Free Energy-Guided Conditional Diffusion Model. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

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

Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. *arXiv preprint arXiv:2407.01521*, 2024.

Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 2016.

APPENDICES

A PROOFS

A.1 PROOF OF PROPOSITION 1

To begin, recall that $p_{0|S}(\mathbf{x}_0|\mathbf{s}) = \frac{1}{Z} p_0(\mathbf{x}_0) e^{\frac{r(\mathbf{x}_0;\mathbf{s})}{\tau}}$. Using Bayes' rule, we can rewrite this expression as

$$\frac{1}{Z} p_0(\mathbf{x}_0) \exp\left(\frac{r(\mathbf{x}_0;\mathbf{s})}{\tau}\right) = \frac{p_{0,S}(\mathbf{x}_0,\mathbf{s})}{p_S(\mathbf{s})} = \frac{p_{S|0}(\mathbf{s} \mid \mathbf{x}_0) p_0(\mathbf{x}_0)}{p_S(\mathbf{s})}$$

and we gather that, for $\mathbf{s}$ fixed, $p_{S|0}(\mathbf{s}|\mathbf{x}_0) \propto e^{\frac{r(\mathbf{x}_0;\mathbf{s})}{\tau}}$ where the proportionality is up to constants on independent of $\mathbf{x}_0$. Starting from the LHS of (3), we first apply Bayes' rule, reverse the conditioning,

and introduce a marginalized $\mathbf{x}_0$ to leverage conditional independence. This sequence leads to

$$
\begin{aligned}
p_{t|t+1,Y,S}(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{y}, \mathbf{s}) &= \frac{p_{t,S|t+1,Y}(\mathbf{x}_t, \mathbf{s} \mid \mathbf{x}_{t+1}, \mathbf{y})}{p_{S|t+1,Y}(\mathbf{s} \mid \mathbf{x}_{t+1}, \mathbf{y})} \\
&= \frac{p_{t|t+1,Y}(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{y}) p_{S|t,t+1,Y}(\mathbf{s} \mid \mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{y})}{p_{S|t+1,Y}(\mathbf{s} \mid \mathbf{x}_{t+1}, \mathbf{y})} \\
&\propto p_{t|t+1,Y}(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{y}) \int_{\mathbf{x}_0} p_{S|t,t+1,Y}(\mathbf{s} \mid \mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{y}) p_{0|t,t+1,Y}(\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{y}) d\mathbf{x}_0 \\
&= p_{t|t+1,Y}(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{y}) \int_{\mathbf{x}_0} p_{S|0,t,t+1,Y}(\mathbf{s} \mid \mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{y}) p_{0|t,Y}(\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{y}) d\mathbf{x}_0 \\
&\propto p_{t|t+1,Y}(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{y}) \int_{\mathbf{x}_0} p_{S|0}(\mathbf{s} \mid \mathbf{x}_0) p_{0|t,Y}(\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{y}) d\mathbf{x}_0 \\
&= p_{t|t+1,Y}(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{y}) \mathbb{E}_{\mathbf{x}_0 \sim p_{0|t,Y}(\cdot|\mathbf{x}_t, \mathbf{y})}[\exp(r(\mathbf{x}_0; \mathbf{s})/\tau)] \\
&\propto p_{t|t+1,Y}(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{y}) \exp(V_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y})).
\end{aligned}
$$

The penultimate step follows from the discussion at the onset of the proof. The last step captures the definition for $V_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y})$ found in Proposition 1. The proof of (4) is similar without conditioning on $t+1$.

Proof of Value-titled KL. Given a distribution p_0 over $\mathbb{R}^d$, a reward function $r : \mathbb{R}^d \to \mathbb{R}$, and $\tau > 0$, we are interested in sampling from the distribution p^* given by

$$
\begin{aligned}
p^* &= \arg\max_p \mathbb{E}_{\mathbf{x}\sim p}[r(\mathbf{x})] - \tau D_{\mathrm{KL}}(p \| p_0) \\
&= \arg\max_p \mathbb{E}_{\mathbf{x}\sim p}\left[r(\mathbf{x}) - \tau \log \frac{p(\mathbf{x})}{p_0(\mathbf{x})}\right] \\
&= \arg\min_p \mathbb{E}_{\mathbf{x}\sim p}\left[\log \frac{p(\mathbf{x})}{p_0(x)} - \frac{r(\mathbf{x})}{\tau}\right] \\
&= \arg\min_p \mathbb{E}_{x\sim p}\left[\log \frac{p(\mathbf{x})}{p_0(\mathbf{x}) e^{r(\mathbf{x})/\tau}}\right] \triangleq \arg\min_p \mathcal{L}(p).
\end{aligned}
$$

Let $q(\mathbf{x}) \triangleq \frac{1}{Z} p_0(\mathbf{x}) e^{r(\mathbf{x})/\tau}$, where Z is chosen such that $\int q(\mathbf{x}) d\mathbf{x} = 1$. Then

$$
\mathcal{L}(p) = \mathbb{E}_{\mathbf{x}\sim p}\left[\log \frac{p(\mathbf{x})}{p_0(\mathbf{x}) e^{r(\mathbf{x})/\tau}}\right] = \mathbb{E}_{\mathbf{x}\sim p}\left[\log \frac{p(\mathbf{x})}{Z q(\mathbf{x})}\right] = D_{\mathrm{KL}}(p \| q) - \log Z.
$$

By the non-negativity of KL-divergence, $\mathcal{L}(p) \geq \mathcal{L}(q)$ for any distribution p, and so $p^* = q$, or $p(\mathbf{x}) \propto p_0(\mathbf{x}) e^{r(\mathbf{x})/\tau}$.

A.2 Value Approximation Bound

In the following, we provide the steps that lead to Eq. 7, and subsequently bound the approximation error. We begin with the following lemma.

Lemma 2. *The conditional mean of X_0 given $X_t = \mathbf{x}_t$ and $Y = \mathbf{y}$ is given by*

$$
\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t, \mathbf{y}) = \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t) + \left(\frac{1 - \alpha_t}{\sqrt{\alpha_t}}\right) \nabla_{\mathbf{x}_t} \log p_{Y|t}(\mathbf{y} \mid \mathbf{x}_t). \tag{8}
$$

Proof. For any distribution over X_0 since $p_{t|0}(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t \mid \sqrt{\alpha_t}\mathbf{x}_0, (1 - \alpha_t)\mathbf{I})$, we can use the Tweedie's formula (Efron, 2011) to $p_{0|Y}(\mathbf{x}_0 \mid \mathbf{y})$ and $p_0(\mathbf{x}_0)$ to get

$$
\sqrt{\alpha_t}\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t, \mathbf{y}) = \mathbf{x}_t + (1 - \alpha_t)\nabla_{\mathbf{x}_t} \log p_{t|Y}(\mathbf{x}_t \mid \mathbf{y}) \tag{9}
$$

$$
\sqrt{\alpha_t}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t) = \mathbf{x}_t + (1 - \alpha_t)\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t). \tag{10}
$$

Since by Bayes theorem, $\nabla_{\mathbf{x}_t} \log p_{t|Y}(\mathbf{x}_t \mid \mathbf{y}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \log p_{Y|t}(\mathbf{y} \mid \mathbf{x}_t)$, the results follows by simple algebra.

Since $p_{Y|0}$ is Gaussian, using the DPS approximation on the second term in Eq. (8), we get

$$\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t, \mathbf{y}) \approx \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t) - \left(\frac{1-\alpha_t}{\sqrt{\alpha_t}}\right) \frac{1}{2\sigma_y^2} \nabla_{\mathbf{x}_t} \|\mathbf{y} - \mathbf{A}(\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))\|_2^2. \tag{11}$$

Replacing $1/2\sigma_y^2$ by η to control the approximation error gives Eq. (6). In the following, we denote this approximation as

$$\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t, \mathbf{y}) \approx \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t) - \left(\frac{1-\alpha_t}{\sqrt{\alpha_t}}\right) \eta \nabla_{\mathbf{x}_t} \|\mathbf{y} - \mathbf{A}(\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))\|_2^2 \triangleq \tilde{\mathbf{x}}_{0|t}^{\eta}(\mathbf{x}_t, \mathbf{y}). \tag{12}$$

Proposition 3. *Assume that r is a Lipschitz function that takes values in $[0, 1]$. For any $\mathbf{x}_t, \mathbf{y}, \mathbf{s}$, the error in the value approximation $\hat{V}_t^{\tau}(\mathbf{x}_t; \mathbf{s}, \mathbf{y}) = r(\tilde{\mathbf{x}}_{0|t}^{\eta}(\mathbf{x}_t, \mathbf{y}); \mathbf{s})/\tau$ with the true value $V_t^{\tau}(\mathbf{x}_t; \mathbf{s}, \mathbf{y})$ is bounded as*

$$|V_t^{\tau}(\mathbf{x}_t; \mathbf{s}, \mathbf{y}) - \hat{V}_t^{\tau}(\mathbf{x}_t; \mathbf{s}, \mathbf{y})| \leq c_\tau c_1(t) + \frac{L_r}{\tau}\left(\sqrt{c_2(t)} + \sqrt{c_3(t) c_4^{\eta}(t)}\right), \tag{13}$$

where $c_1(t) = \mathrm{Var}\left(r(X_0; \mathbf{s}) \mid \mathbf{x}_t, \mathbf{y}\right)$, $c_2(t) = \mathrm{Var}(X_0 \mid \mathbf{x}_t, \mathbf{y})$, $c_3(t) = \mathrm{Var}(X_0 \mid \mathbf{x}_t)$,

$$c_4^{\eta}(t) = 1 + \mathrm{CV}^2(t) + \eta^2 \|\mathbf{A}^T(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))\|^2_{\Sigma_{0|t}(\mathbf{x}_t)} \tag{14}$$

$$- 2\eta \langle \mathbf{A}(\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t, \mathbf{y}) - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)), \mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)\rangle, \tag{15}$$

where $\mathrm{CV}(t) \triangleq \frac{\sqrt{\mathrm{Var}(p_{Y|0}(\mathbf{y}|X_0)|\mathbf{x}_t)}}{\mathbb{E}[p_{Y|0}(\mathbf{y}|X_0)|\mathbf{x}_t]}$ is the coefficient of variation of the likelihood function $p_{Y|0}(\mathbf{y} \mid X_0)$ given $\mathbf{x}_t$ and $c_\tau = e^{1/\tau} - 1 - 1/\tau$ is a positive constant.

Remark 4. *In Proposition 3, the term $c_1(t)$ denotes the conditional variance of the reward given $\mathbf{x}_t, \mathbf{y}$, $c_2(t)$ denotes the conditional variance of X_0 given $\mathbf{x}_t, \mathbf{y}$, and $c_3(t)$ denotes the conditional variance of X_0 given only $\mathbf{x}_t$. Since the variance of the reverse distribution $p_{0|t}(\cdot \mid \mathbf{x}_t)$ decreases as t becomes smaller, we have that all the terms $c_1(t), c_2(t), c_3(t)$ are small when t is small. Therefore, the approximation error is small when t is small.*

Proof. Since r is a bounded random variable, assuming finite variance, we can use Bennett's inequality for the log moment-generating function

$$V_t^{\tau}(\mathbf{x}_t; \mathbf{s}, \mathbf{y}) = \log \mathbb{E}[\exp(r(X_0; \mathbf{s})/\tau)] \leq \frac{1}{\tau}\mathbb{E}[r(X_0; \mathbf{s})] + c_\tau c_1(t). \tag{16}$$

Then, we have

$$|\log \mathbb{E}[\exp(r(X_0; \mathbf{s})/\tau)] - \frac{1}{\tau} r(\tilde{\mathbf{x}}_{0|t,Y}^{\eta}(\mathbf{x}_t, \mathbf{y}))| \leq \frac{1}{\tau}|\mathbb{E}[r(X_0; \mathbf{s})] - r(\tilde{\mathbf{x}}_{0|t,Y}^{\eta}(\mathbf{x}_t, \mathbf{y}))| + c_\tau c_1(t). \tag{17}$$

Now, let us simplify the first term,

$$|\mathbb{E}[r(X_0; \mathbf{s})] - r(\tilde{\mathbf{x}}_{0|t,Y}^{\eta}(\mathbf{x}_t, \mathbf{y}); \mathbf{s})| \leq \mathbb{E}[|r(X_0; \mathbf{s}) - r(\tilde{\mathbf{x}}_{0|t,Y}^{\eta}(\mathbf{x}_t, \mathbf{y}); \mathbf{s})|] \tag{18}$$

$$\leq L_r \mathbb{E}[\|X_0 - \tilde{\mathbf{x}}_{0|t,Y}^{\eta}(\mathbf{x}_t, \mathbf{y})\|_2] \tag{19}$$

$$\leq L_r(\mathbb{E}[\|X_0 - \hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t, \mathbf{y})\|_2] + \|\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t, \mathbf{y}) - \tilde{\mathbf{x}}_{0|t,Y}^{\eta}(\mathbf{x}_t, \mathbf{y})\|_2). \tag{20}$$

The first term can be bounded by $\sqrt{c_2(t)}$ using Cauchy-Schwarz inequality in L^2-probability space. For the second term, first we simplify

$$\tilde{\mathbf{x}}_{0|t,Y}^{\eta}(\mathbf{x}_t, \mathbf{y}) = \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t) - \left(\frac{1-\alpha_t}{\sqrt{\alpha_t}}\right) \eta \nabla_{\mathbf{x}_t} \|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)\|_2^2,$$

and then $\nabla_{\mathbf{x}_t} \|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)\|_2^2 = -(\nabla_{\mathbf{x}_t} \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))\mathbf{A}^T(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))$. Now,

$$\nabla_{\mathbf{x}_t} \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \int \mathbf{x}_0^T p_{0|t}(\mathbf{x}_0 \mid \mathbf{x}_t) \mathrm{d}\mathbf{x}_0 = \int \nabla_{\mathbf{x}_t} p_{0|t}(\mathbf{x}_0 \mid \mathbf{x}_t) \mathbf{x}_0^T \mathrm{d}\mathbf{x}_0 \tag{21}$$

$$= \int \nabla_{\mathbf{x}_t} \log p_{0|t}(\mathbf{x}_0 \mid \mathbf{x}_t) \mathbf{x}_0^T p_{0|t}(\mathbf{x}_0 \mid \mathbf{x}_t) \mathrm{d}\mathbf{x}_0. \tag{22}$$

Now, we shall compute

$$\nabla_{\mathbf{x}_t} \log p_{0|t}(\mathbf{x}_0 \mid \mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log p_{t|0}(\mathbf{x}_t \mid \mathbf{x}_0) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t). \tag{23}$$

But since $\sqrt{\alpha_t}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t) = \mathbf{x}_t + (1-\alpha_t)\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$, and $\nabla_{\mathbf{x}_t} \log p_{t|0}(\mathbf{x}_t \mid \mathbf{x}_0) = \frac{1}{1-\alpha_t}(\sqrt{\alpha_t}\mathbf{x}_0 - \mathbf{x}_t)$, which gives

$$\left(\frac{1-\alpha_t}{\sqrt{\alpha_t}}\right) \nabla_{\mathbf{x}_t} \log p_{0|t}(\mathbf{x}_0 \mid \mathbf{x}_t) = \mathbf{x}_0 - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t), \tag{24}$$

which gives

$$\left(\frac{1-\alpha_t}{\sqrt{\alpha_t}}\right) \nabla_{\mathbf{x}_t} \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t) = \mathbb{E}_{X_0 \sim p_{0|t}(\mathbf{x}_t)}[(X_0 - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))X_0^T] \tag{25}$$

$$= \mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[X_0 X_0^T] - \mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[X_0]\mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[X_0^T] \tag{26}$$

$$= \mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[(X_0 - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))(X_0 - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))^T] \triangleq \Sigma_{0|t}(\mathbf{x}_t), \tag{27}$$

which is precisely the covariance matrix of X_0 given $\mathbf{x}_t$.

Thus, we finally get,

$$\tilde{\mathbf{x}}^{\eta}_{0|t,Y}(\mathbf{x}_t, \mathbf{y}) = \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t) + \eta \Sigma_{0|t}(\mathbf{x}_t)\mathbf{A}^T(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)) \tag{28}$$

Next, note that since $\mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[p_{Y|0}(\mathbf{y} \mid X_0)] = p_{Y|t}(\mathbf{y} \mid \mathbf{x}_t)$, we can define $f(X_0) = \frac{p_{Y|0}(\mathbf{y}|X_0)}{\mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[p_{Y|0}(\mathbf{y}|X_0)]}$, whose expectation is $\mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[f(X_0)] = 1$. Further, it is easy to see that $\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t, \mathbf{y}) = \mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[X_0 f(X_0)]$. Now, we are ready to bound the final term as follows

$$\|\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t, \mathbf{y}) - \tilde{\mathbf{x}}^{\eta}_{0|t,Y}(\mathbf{x}_t, \mathbf{y})\|_2 \tag{29}$$

$$= \|\mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[X_0 f(X_0)] - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t) - \eta\Sigma_{0|t}(\mathbf{x}_t)\mathbf{A}^T(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))\|_2 \tag{30}$$

$$= \|\mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[X_0 f(X_0) - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t) f(X_0)] \tag{31}$$

$$- \eta\mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[(X_0 - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))(X_0 - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))^T]\mathbf{A}^T(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))\|_2 \tag{32}$$

$$= \|\mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[(X_0 - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))(f(X_0) - \eta(X_0 - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))^T\mathbf{A}^T(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))))]\|_2 \tag{33}$$

$$\leq \mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[\|(X_0 - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))(f(X_0) - \eta(X_0 - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))^T\mathbf{A}^T(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))))\|_2] \tag{34}$$

$$= \mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[\|X_0 - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)\|_2 |f(X_0) - \eta(X_0 - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))^T\mathbf{A}^T(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))|] \tag{35}$$

$$\leq \sqrt{c_3(t)}\sqrt{c_4^{\eta}(t)}, \tag{36}$$

where the last step follows by Cauchy-Schwarz inequality in L^2 probability space, where

$$c_3(t) \triangleq \mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[\|X_0 - \mathbb{E}_{p_{0|t}(\mathbf{x}_t)[X_0]}\|_2^2] = \mathrm{Var}(X_0 \mid \mathbf{x}_t). \tag{37}$$

and at last, we have

$$c_4^{\eta}(t) \triangleq \mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[(f(X_0) - \eta(X_0 - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))^T\mathbf{A}^T(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)))^2] \tag{38}$$

$$= \mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[f(X_0)^2] + \eta^2\mathbb{E}[((X_0 - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))^T\mathbf{A}^T(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)))^2] \tag{39}$$

$$- 2\eta\mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[(X_0 f(X_0) - f(X_0)\hat{\mathbf{x}}_{0|t}(t))^T]\mathbf{A}^T(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)) \tag{40}$$

$$= \mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[f(X_0)^2] + \eta^2\|\mathbf{A}^T(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))\|^2_{\Sigma_{0|t}(\mathbf{x}_t)} \tag{41}$$

$$- 2\eta\langle\mathbf{A}(\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t, \mathbf{y}) - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)), \mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)\rangle. \tag{42}$$

Since $\mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[f(X_0)^2] = \frac{\mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[(p_{Y|0}(\mathbf{y}|X_0))^2]}{(\mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[p_{Y|0}(\mathbf{y}|X_0)])^2} = 1 + \mathrm{CV}^2(t)$, where $\mathrm{CV}(t) = \frac{\sqrt{\mathrm{Var}(p_{Y|0}(\mathbf{y}|X_0)|\mathbf{x}_t)}}{\mathbb{E}[p_{Y|0}(\mathbf{y}|X_0)|\mathbf{x}_t]}$ is the coefficient of variation of the likelihood function given $\mathbf{x}_t$.

Algorithm 1 Inference-Time Search with Side Information for Inverse Problems

Require: Side information $\mathbf{s}$, observation $\mathbf{y}$, reward function r, resampling parameter B, number of particles N, temperature $\tau > 0$

1: Initialize N particles: $\mathbf{x}_T[i] \sim \mathcal{N}(0, \mathbf{I})$ for $1 \leq i \leq N$
2: **for** $t = T - 1$ to 0 **do**
3: Sample $\mathbf{x}_t[i] \sim p_{t|t+1,Y}(\cdot \mid \mathbf{x}_{t+1}[i], \mathbf{y})$, ▷ Sample candidate particles
4: $r[i] \leftarrow r(\hat{\mathbf{x}}_{0|t,Y}[i]; \mathbf{s})$ ▷ Compute reward using side information
5: $g_t \leftarrow \textsc{GroupSize}(N, B, t)$ ▷ Compute the group size at step t for resampling
6: $I \leftarrow \textsc{Resample}(r, g_t, \tau)$ ▷ Resample indices with replacement among the groups of size g_t
7: $\mathbf{x}_t[i] \leftarrow \mathbf{x}_t[I[i]]$ ▷ Retain the particles in the resampled indices
8: Select $\mathbf{x}_0^*$ from $\hat{\mathbf{x}}_{0|0}[1:N]$ (e.g., via reward maximization) **return** $\mathbf{x}_0^*$

A.3 Algorithm

We summarize our framework in Algorithm 1. The GroupSize step in line 5 computes the group-size at time t and can be changed to obtain various search strategies: Best-of-N, Greedy Search, and RFJ Search, and the Resample step in line 6 samples the indices within the groups as described in the main paper. Line 3 is specific to the diffusion samplers and how they implement it.

Here, we roughly explain how the entire algorithm is implemented in all the three diffusion samplers.

DPS: Compute the denoised mean and the clean data estimate $\boldsymbol{\mu}_t[i], \hat{\mathbf{x}}_{0|t}[i]$ from $\mathbf{x}_{t+1}[i]$. Compute $\mathbf{g}_t[i] = \nabla_{\mathbf{x}_t} \|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_{t+1}[i])\|_2^2$ and use it to update $\hat{\mathbf{x}}_{0|t,Y}[i] \leftarrow \hat{\mathbf{x}}_{0|t}[i] - \frac{1-\alpha_t}{\sqrt{\alpha_t}}\eta \mathbf{g}_t[i]$, and $\boldsymbol{\mu}_t[i] \leftarrow \boldsymbol{\mu}_t[i] - \zeta \mathbf{g}_t[i]$. Then, compute rewards based on $\hat{\mathbf{x}}_{0|t,Y}[i]$ to resample promising indices. Take the reverse diffusion step on the resampled $\boldsymbol{\mu}_t[I[i]]$ to obtain $\mathbf{x}_t[i]$.

DAPS: Compute the clean data estimate $\hat{\mathbf{x}}_{0|t}[i]$ from $\mathbf{x}_{t+1}[i]$. Compute the rewards based on $\hat{\mathbf{x}}_{0|t}[i]$, resample, and then take MCMC steps, starting from the resampled particles, to perform a local Langevin sampling (Zhang et al., 2024). In the end, we obtain $\hat{\mathbf{x}}_{t,Y}[i]$, from which we sample $\mathbf{x}_t[i]$ by adding appropriate decoupled noise (Zhang et al., 2024).

MPGD: Compute the clean data estimate $\hat{\mathbf{x}}_{0|t}[i]$ from $\mathbf{x}_{t+1}[i]$. Compute gradient $\mathbf{g}_t = \nabla_{\mathbf{x}_{0|t}} \|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{0|t}[i]\|_2^2$. Take $\hat{\mathbf{x}}_{0|t,Y}[i] \leftarrow \hat{\mathbf{x}}_{0|t}[i] - \frac{1-\alpha_t}{\sqrt{\alpha_t}}\eta \mathbf{g}_t[i]$, and $\hat{\mathbf{x}}_{0|t}[i] \leftarrow \hat{\mathbf{x}}_{0|t}[i] - \zeta \mathbf{g}_t$. Resample indices based on rewards computed from $\hat{\mathbf{x}}_{0|t,Y}[i]$. Then, using the particles corresponding to the sampled indices $\hat{\mathbf{x}}_{0|t,Y}[I[i]]$, take a reverse DDIM step (He et al., 2024; Song et al., 2021a).

B Additional Experiments

B.1 DPS

Figure 7 provides a qualitative example highlighting why FaceSimilarity is essential in addition to standard metrics such as PSNR, SSIM, and LPIPS. It compares three samples generated by the BlindDPS algorithm with our RFJS reconstruction. Although our result better preserves identity and achieves a higher FaceSimilarity score, its PSNR and LPIPS are worse than BlindDPS, underscoring that traditional metrics may not capture semantic fidelity. This illustrates the need for task-specific metrics in inverse problems to ensure qualitative performance.

We also present additional qualitative examples for our DPS experiments in Figures 8 and 9, corresponding to settings that use face and text as side information, respectively.

B.2 DAPS

Setup: We employ the search and gradient modules to infuse side information using DAPS as the base sampler. We consider two challenging tasks, box inpainting with a box of size 96×96 and super-resolution with downsampling factor of 10. For gradient guidance, we used a scale of 15 with respect to the noise being added to $\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)$ after MCMC steps (Zhang et al., 2024). DAPS uses fewer diffusion steps (200) than DPS (1000). Further, the algorithm is based on a graphical model that allows for more inherent exploration due to the decoupling between consecutive steps. Therefore, for search algorithms, relatively smaller bases are preferable, and hence $B = 4$ is chosen.

Figure 7: Qualitative illustration of the relevance of the FaceSimilarity metric and the superior performance of RFJS in identity preservation. RFJS reconstruction is clearly more faithful to the ground truth, yet PSNR, SSIM, and LPIPS values slightly favor the BlindDPS outputs.

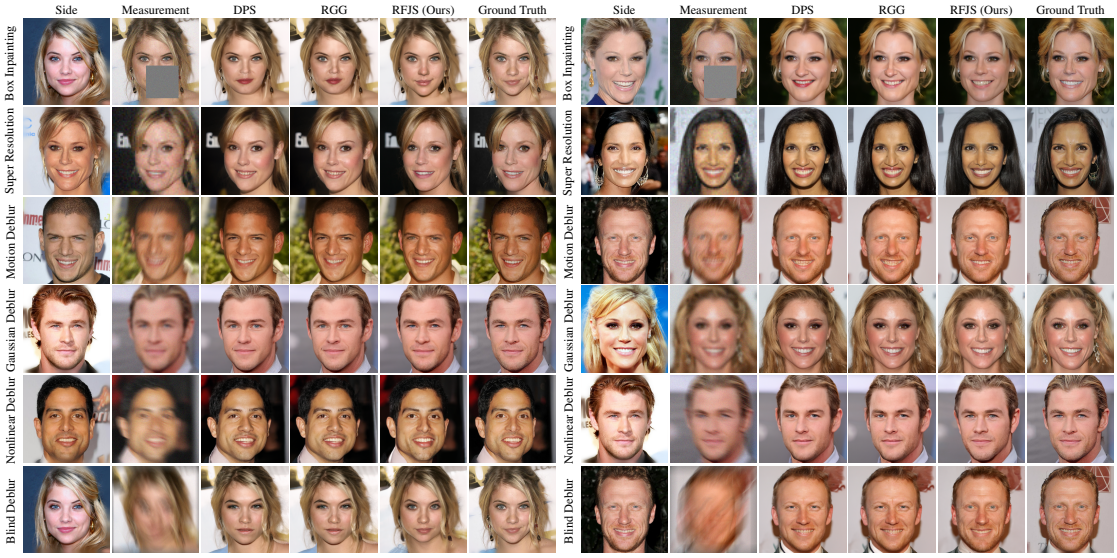


Figure 8: Additional samples using DPS as the base sampler.

Results: The qualitative results are given in Figure 10, while the quantitative metrics are given in 3. Observe that for the task of inpainting, our search algorithm shows a significant improvement over DAPS, particularly in the FaceSimilarity metric.

	Box Inpainting				Super Resolution (×10)			
Method	FaceSimilarity (↓)	PSNR (↑)	LPIPS (↓)	SSIM (↑)	FaceSimilarity (↓)	PSNR (↑)	LPIPS (↓)	SSIM (↑)
RFJS (ours)	**0.423**$\pm$0.10	**28.720**$\pm$1.35	**0.140**$\pm$0.03	**0.788**$\pm$0.03	<u>0.654</u>$\pm$0.11	<u>25.228</u>$\pm$1.34	**0.282**$\pm$0.03	<u>0.661</u>$\pm$0.04
GS (ours)	0.511$\pm$0.12	28.640$\pm$1.43	<u>0.140</u>$\pm$0.03	<u>0.787</u>$\pm$0.03	0.760$\pm$0.12	**25.271**$\pm$1.36	0.285$\pm$0.03	**0.662**$\pm$0.04
RGG	<u>0.436</u>$\pm$0.12	28.410$\pm$1.39	0.141$\pm$0.03	0.784$\pm$0.03	**0.579**$\pm$0.13	25.210$\pm$1.34	<u>0.282</u>$\pm$0.03	0.659$\pm$0.04
BON	0.611$\pm$0.14	<u>28.660</u>$\pm$1.45	0.141$\pm$0.03	0.787$\pm$0.03	0.909$\pm$0.11	25.220$\pm$1.38	0.285$\pm$0.03	0.660$\pm$0.04
DAPS	0.739$\pm$0.18	28.290$\pm$1.53	0.142$\pm$0.03	0.784$\pm$0.03	1.020$\pm$0.14	25.170$\pm$1.35	0.285$\pm$0.03	0.659$\pm$0.04

Table 3: Comparison of metrics for various inverse problems using DAPS as the base sampler. For each metric, the best result is shown in **bold**, and the second best is <u>underlined</u>. We observe that our RFJ Search-based algorithm has the best or the second-best performance in all the tasks.

B.3 MPGD

Setup: As in the paper He et al. (2024), we choose super-resolution and Gaussian deblurring as the tasks, along with additional task of box inpainting. For box inpainting, we used a box of size 64×64 at the center of the face. Further, we modified the down-sampling scale of super-resolution from 4 to 6, and the intensity of the kernel in Gaussian deblur from 3 to 5 to make the tasks more challenging. Even though MPGD uses 100 DDIM steps in generation, its exploratory capabilities are similar to DPS. Therefore, we cannot use very large base B, whence, we choose $B = 8$ for box inpainting, and super-resolution. For Gaussian deblur, we found that using $B = 8$ becomes detrimental for other metrics, and so $B = 16$ is used. For the gradient guidance, the scales of 0.5 for box inpainting, and

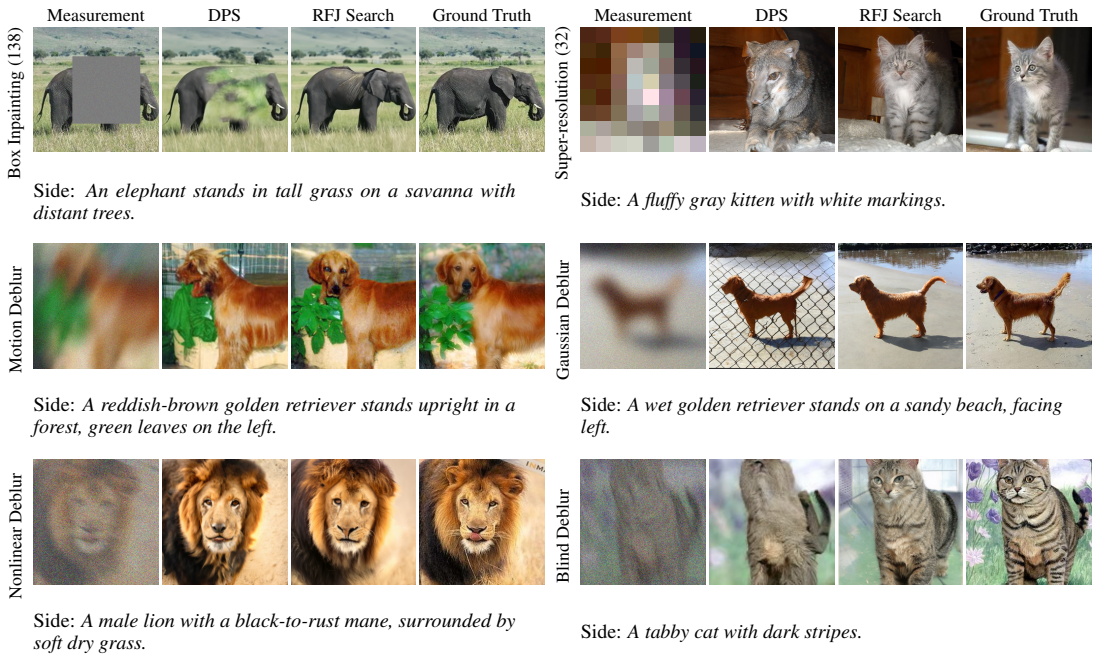


Figure 9: Qualitative comparison on ImageNet with textual side information. For highly degraded observations, DPS and BlindDPS often produce artifacts, whereas our method reduces these by better aligning reconstructions with the description.

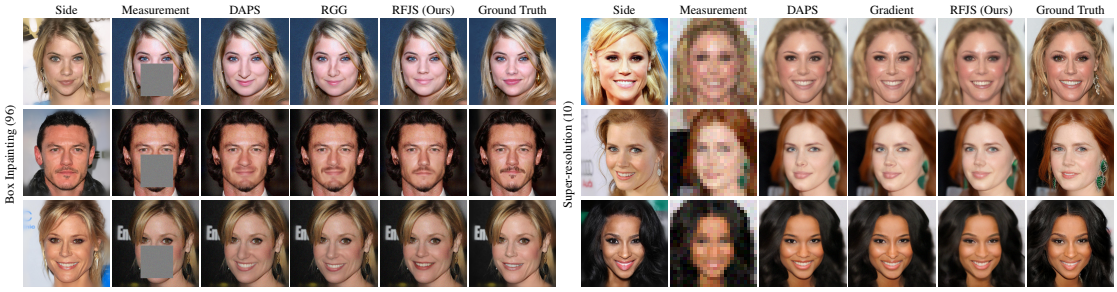


Figure 10: Qualitative comparison of algorithms using DAPS as the base sampler. Our method offers better reconstructions aligned with the identity.

0.25 for super-resolution and Gaussian deblur are chosen carefully to avoid overfitting. This is the scale relative to the gradient with respect to the measurement. For more discussion and examples on the effect of gradient scale, see Appendix B.4.

Results: Experimental results with MPGD (He et al., 2024) as baseline algorithm are given in Table 4. We observe that using RFJ Search significantly enhances the FaceSimilarity (FS) metric, while improving the other metrics. The reconstructions that utilize the side information exhibit strong identity match with the one described by the measurement, which is reflected in the FS metric.

B.4 Experiments on Effect of Gradient Scale

Gradient Guidance Limitations. While guiding the reverse diffusion process with reward gradients can help generate images with higher reward scores, this approach has several limitations. First, as shown in Figures 12, 13, gradient-based guidance primarily adds fine details, such as wrinkles or texture, to the reconstruction, but it cannot significantly alter the global structure of the face. To isolate the effect of the gradient, we used fixed noise realizations for both the gradient-based and baseline methods. The results show that changes are mostly confined to local details, implying that if

Sampler	Task	Method	FaceSimilarity (↓)	PSNR (↑)	LPIPS (↓)	SSIM (↑)
MPGD	Box Inpainting (64)	RFJS (8) (ours)	**0.542**±0.08	**29.81**±1.44	**0.102**±0.02	**0.852**±0.02
		GS (8) (ours)	0.587±0.10	29.44±1.75	0.102±0.02	0.851±0.02
		RGG (0.5)	0.609±0.08	29.24±1.30	0.103±0.02	0.850±0.02
		BON	0.661±0.08	29.35±1.82	0.102±0.02	0.851±0.02
		MPGD	0.766±0.07	29.09±1.27	0.103±0.02	0.848±0.02
	Super Resolution (6)	RFJS (8) (ours)	**0.834**±0.09	**24.50**±1.48	**0.242**±0.04	**0.666**±0.06
		GS (8) (ours)	0.878±0.08	24.45±1.47	0.247±0.03	0.660±0.06
		RGG (0.25)	0.854±0.07	24.39±1.44	0.246±0.03	0.656±0.05
		BON	0.964±0.09	24.44±1.58	0.244±0.04	0.664±0.06
		MPGD	1.037±0.07	24.39±1.45	0.249±0.03	0.657±0.06
	Gaussian Deblur (5)	RFJS (16) (ours)	0.848±0.07	24.19±1.40	**0.229**±0.03	0.638±0.06
		GS (16) (ours)	0.893±0.07	24.14±1.39	0.233±0.03	0.637±0.06
		RGG (0.25)	**0.846**±0.05	24.11±1.34	0.235±0.03	0.634±0.05
		BON	0.950±0.07	**24.20**±1.38	0.233±0.03	**0.640**±0.06
		MPGD	1.026±0.06	24.09±1.35	0.236±0.03	0.634±0.06

Table 4: Quantitative comparison of reconstruction metrics in case of inverse problems with MPGD as the base sampler. For each metric, the best result is shown in **bold**, and the second best is underlined. Observe that our RFJSsearch algorithm has the best or the second-best performance in all the tasks. In Gaussian deblur, our search algorithm is only marginally worse than the best metrics attained. The value in the brackets indicates the resampling rate for search algorithms, and gradient scale for Gradient algorithm.

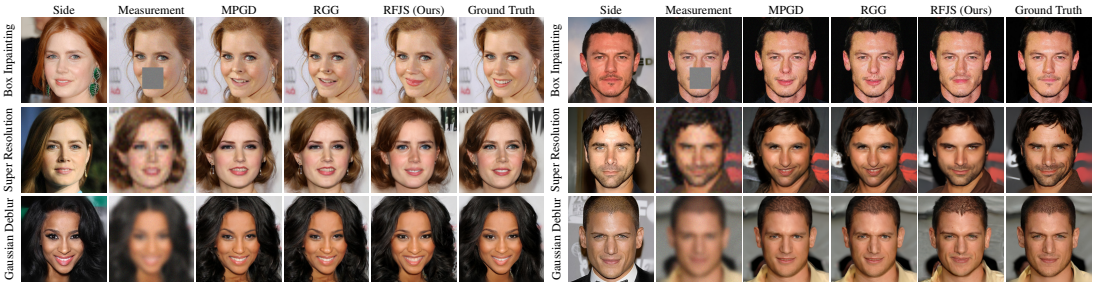


Figure 11: Qualitative comparison of algorithms using MPGD as the base sampler. Our method offers better reconstructions aligned with the identity. Notice that while the identity is preserved, the exact reconstruction might not be possible as witnessed in, for example, Box inpainting. The ground truth has a smiling face whereas the reconstruction does not, although being the same person. Thus, PSNR improvements over the base sampler might be small, but FaceSimilarity improvements are significant.

the sampling trajectory is poor, gradient guidance alone cannot compensate. This highlights the need for search-based methods that can explore a wider range of trajectories during inference.

Second, this method is sensitive to the choice of gradient scale. In the visual examples, we used a relatively large scale of 1.6 to make the gradient's effect more visible; however, such high scales often degrade other metrics like PSNR and SSIM and introduce artifacts. Empirically, we found that a scale around 0.5 yields the best balance when the base sampler is DPS or BlindDPS, consistently improving the FaceSimilarity metric while preserving other evaluation metrics and avoiding artifacts (see Figure 13). Moreover, the sensitivity to gradient scale increases when the number of reverse diffusion steps is small. For instance, in DAPS (Figure 12) and MPGD, where the number of steps is limited to 200 and 100 respectively, larger scales quickly lead to visible artifacts, as demonstrated in Figures 14, and 15.

It is well established in deep learning research that deep (convolutional) neural networks are vulnerable to gradient-based adversarial attacks (Goodfellow et al., 2015). Consequently, using such networks to provide reward-based guidance inherits these vulnerabilities. However, when combined with diffusion samplers, this susceptibility is partially alleviated, as the diffusion process can help steer trajectories away from adversarially induced local minima. This mitigating effect is particularly evident when using a large number of sampling steps (e.g., 1000 steps in DPS). In contrast, samplers

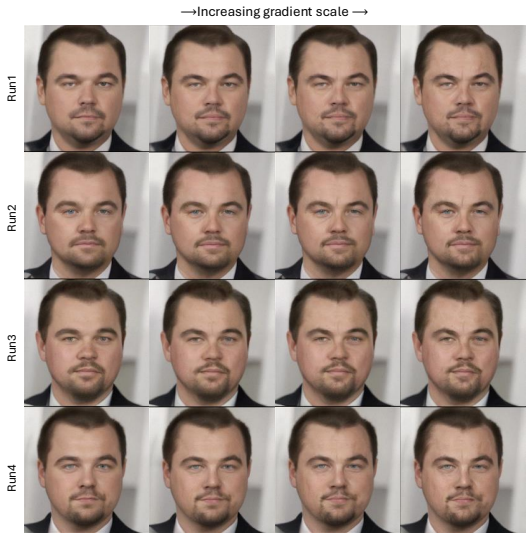


Figure 12: **Effect of reward-gradient guidance in diffusion-based inverse problems.** We show 4 runs with different random seeds (rows), and for each seed we vary the gradient scale across 4 settings (columns). Within each row, the noise realization is identical and only the gradient scale changes; within each column, the gradient scale is fixed while the random seed varies. The ground truth and degraded input are the same for all reconstructions. This arrangement reveals two key observations: (1) The reward gradient influences fine details, such as wrinkles and facial lines without altering the overall facial structure; the structure is primarily determined by the initial noise realization. (2) Different seeds reconstruct different face structures, highlighting the multi-modal nature of the problem. This demonstrates why using multiple particles and performing search across them is beneficial: it enables exploration of structurally different hypotheses while the reward gradient refines locally.

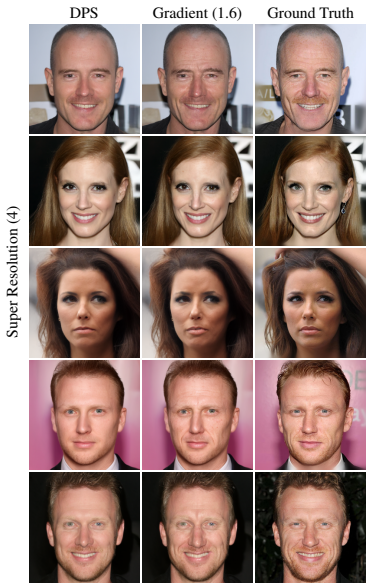


Figure 13: Qualitative comparison of the effect of gradient scale on reconstruction paths.

with fewer steps (e.g., 100 steps in MPGD) exhibit increased sensitivity to the gradient scale, as illustrated in Figures 14 and 15.

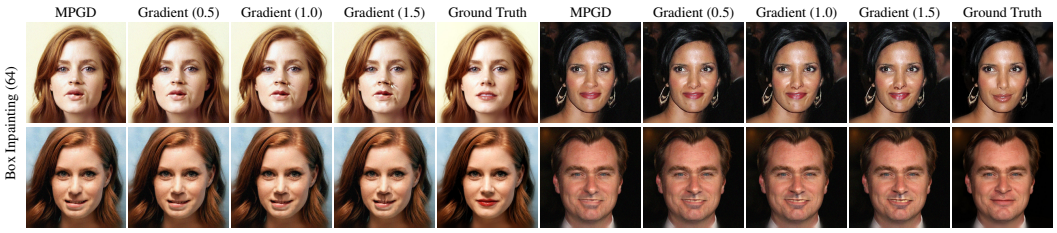


Figure 14: Qualitative comparison of the effect of gradient scale on reconstruction paths. Notice that while the base reconstruction is reasonable, adding the gradient can degrade it if the scale is very large. The final scale used in the experiments is 0.5.

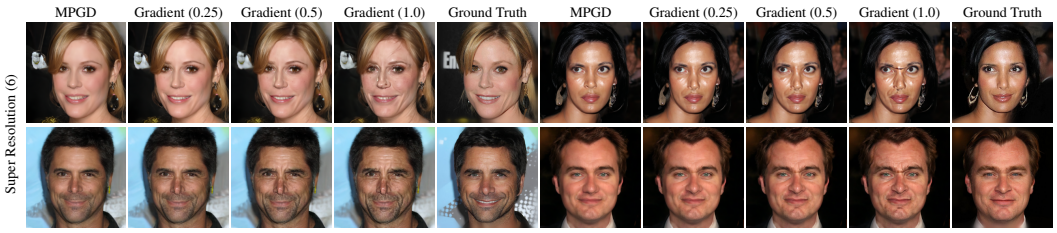


Figure 15: Qualitative comparison of the effect of gradient scale on reconstruction paths for super resolution task. Notice that while the base reconstruction is reasonable, adding the gradient can degrade it if the scale is very large. The final scale used in the experiments is 0.25.

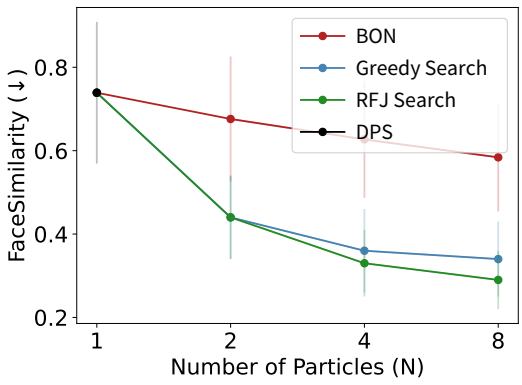
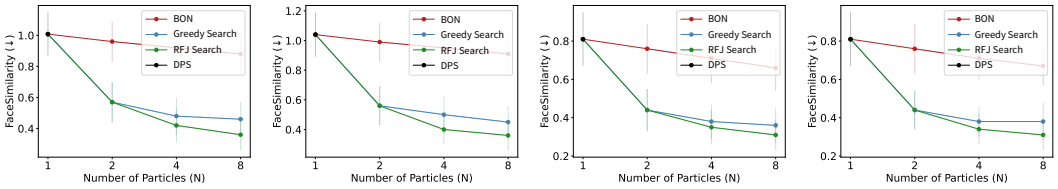
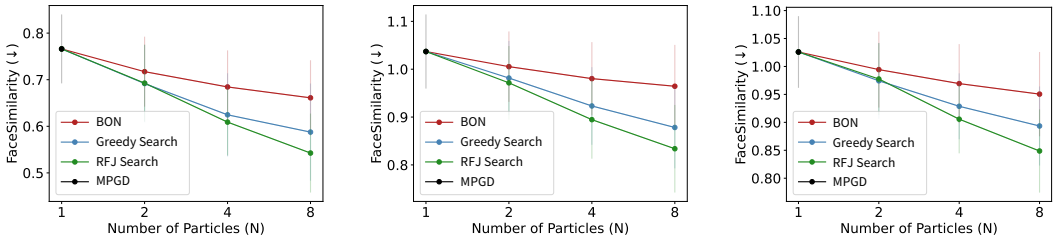


Figure 16: Scaling of search algorithms with respect to the number of particles.



(a) Nonlinear Deblur (b) Super Resolution (4) (c) Gaussian Deblur (d) Motion Deblur

Figure 17: Effect of the number of particles N on the FaceSimilarity metric in DPS. As N increases, we proportionally increase the base to encourage greater exploration. Specifically, for $N = 2, 4, 8$, the corresponding base values are $B = 4, 8, 16$. Using many particles with a small value for base B can overly emphasize reward maximization, which may degrade other evaluation metrics.



(a) Box Inpainting (Box: 64) (b) Super-resolution (Scale: 6) (c) Gaussian Blur (Intensity: 5)

Figure 18: Effect of number of particles N on the FaceSimilarity metric. RFJ Search algorithm offers the best scaling performance, followed by Greedy Search algorithm. Finally, BestOfN performance improves, but only marginally. 64 indicates the size of the box for inpainting, 6 indicates the down-sampling factor in super-resolution, and 5 is the intensity of the Gaussian kernel in Gaussian deblur. The base resampling rate B is 8 for both the search algorithms in box inpainting and super-resolution whereas B is 16 for the Gaussian deblur.

B.5 Effect of Number of Particles

Although we do not explicitly define a second base, the recursive structure of our RFJS algorithm implicitly induces a small base, denoted by b, which refers to the interval at which the smallest groups are resampled. For instance, with 8 particles and a base $B = 16$, the smallest groups of size 2 are resampled every 4 steps, so the induced small base is $b = 4$. This hierarchical resampling schedule enables finer control over exploration without over-committing the entire particle set too aggressively. For analyzing the effect of particle count in RFJ Search (see Figure 17), we fix the small base b so that groups of the same size follow the same resampling dynamics, even across experiments with different total numbers of particles. This ensures that any differences we observe are attributable to the number of particles, not to changes in resampling frequency. On the other hand, when comparing RFJ Search vs Greedy Search, we fix the big base B to ensure both methods resample at the same overall frequency. This is why, in our plots, the B values for Greedy search vary with the number of particles (e.g., $B = 4, 8, 16$ for $2, 4, 8$ particles respectively), matching the corresponding configurations in RFJ Search.

B.6 Runtimes

Table 5 reports wall-clock runtimes (in seconds) for our search algorithms compared to Best-of-N (BON) and Greedy Search across different baselines (DPS, DAPS, and MPGD) with varying numbers of particles. When the number of particles is set to 1, the runtime corresponds to the baseline method without search. As the number of particles increases, amount of computation scales linearly with N, but thanks to parallelization, the wall-clock overhead remains moderate: with $N = 8$, runtimes are only 4–5$\times$ those of the baseline. We also note that our RFJ Search method take slightly more time than BON and Greedy Search, but consistently achieve better reconstruction quality, highlighting the practical efficiency of our approach.

Particles	BON	Greedy Search	RFJ Search
1	55	-	-
2	65	75	75
4	102	118	131
8	180	195	241

(a) Task: Box inpainting. Baseline: DPS

Particles	BON	Greedy Search	RFJS
1	61	-	-
2	72	91	91
4	125	141	157
8	229	245	290

(b) Task: Box inpainting. Baseline: DAPS

Particles	BON	Greedy Search	RFJS
1	3	-	-
2	4	5	5
4	5	8	9
8	8	12	23

(c) Task: Box inpainting. Baseline: MPGD

Table 5: Runtime (seconds) vs number of particles for BON, Greedy Search, and RFJ Search methods on DPS, DAPS and MPGD ($B = 8$). The baseline algorithm corresponds to $N = 1$.

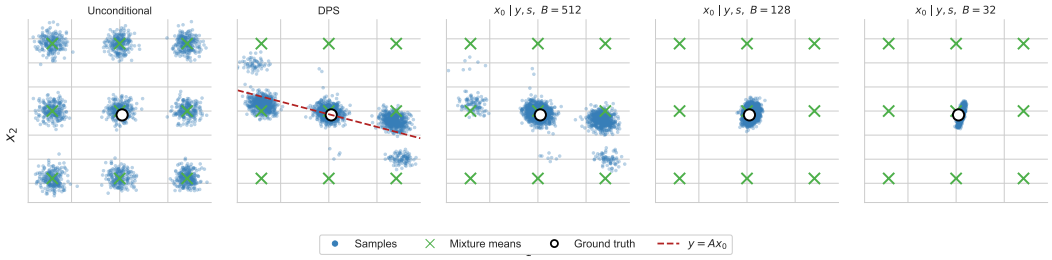


Figure 19: Illustration of the effect of B in utilization of side information for the reconstruction in a linear inverse problem with a mixture of Gaussian prior.

B.7 Simulation in 2D

To understand the effect of side-information in inference-time search, we consider a simple scenario with the prior p_0 on X_0 being a mixture of Gaussian distribution, as illustrated in Figure 19, part (i) (numbered left to right).

For the ground truth $\mathbf{x}$ sampled from the central mode, and given $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}_y$, we plotted the posterior estimates by the DPS algorithm in (ii). Notice that since the posterior is multi-modal with three modes due to the prior distribution, the reconstruction algorithm DPS cannot recover the ground truth with accuracy. Given additional side information, which has a linear relation $\mathbf{s} = \mathbf{B}\mathbf{x} + \mathbf{z}_s$ with $\mathbf{x}$ (vertical, not plotted), we plot the samples generated by using our RFJS algorithm as the base rate B decreases as $512, 128, 32$ in Figures (iii), (iv) and (v) respectively. We use the reward function $r(\mathbf{x}, \mathbf{s}) = -\|\mathbf{s} - \mathbf{B}\mathbf{x}\|_2^2$. As B decreases, the effect of side information is more visible as the generated samples concentrate around the ground truth $\mathbf{x}$. This illustrates the importance and usefulness of side information in a simple yet intuitive setting.

The above example considers a linear side information with negative of the distance in the range space of $\mathbf{B}$, which is a concave and hence well-behaved reward. Therefore, B has more or less monotone effect on the performance of the RFJS, the particles get closer and closer to the ground truth. We wish to illustrate the effect of B when $\mathbf{s}$ and $\mathbf{x}$ have a complicated non-linear relationship, which can be represented by a neural network r_θ. In that case, the reward is not very well-behaved. We examine the performances of RJFS and GS algorithms in this setup. We emphasize that this setup accurately captures the nuances in the real-world problems, where the reward model is a pretrained network (for example, in our experiments on face and text as side information). We initialize the reward network with random weights, and we generate samples using this reward for tilting. We compute the PSNR of the reconstucted samples with respect to the ground truth. We plot the result in Figure 20. Observe that RFJS outperforms GS across all values of B, showing the robust nature of RFJS when the reward is complicated non-linear, non-convex function. Finally, it is interesting to observe that the optimal value of PSNR occurs at $B = 4$ using RFJS, indicating that delicate balance is required while utilizing side information.

C Hyperparameters

In our implementation of Resample in Algorithm 1, we perform a greedy resampling, i.e., we pick the best candidate within each group of size g_t and replicate it g_t times. Since we are using large enough B, this is justified and has similar effect as using smaller B with moderate temperature, with the added advantage of utilizing less function calls to the reward network r. Thus, tuning the B allows us to maintain balance without over-optimizing with respect to the reward.

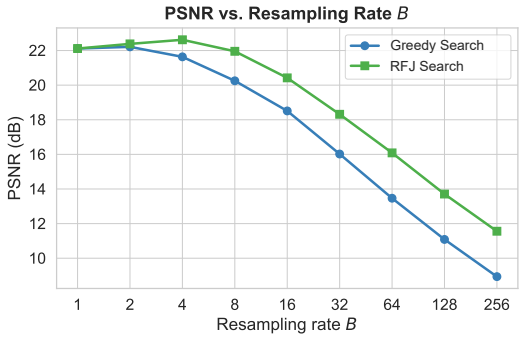


Figure 20: Comparison of performance of RFJ Search (RFJS) and Greedy Search (GS) as a function of B for a randomly generated reward network r_θ. RFJS outperforms GS across all values of B.

For reproducibility, we provide detailed settings for each task, sampler, and search algorithm used in our experiments below.

C.1 Face Identity Experiments

For experiments with face identity as side information:

- **DPS:** Box inpainting with box size 96. All other parameters (downsampling rate, blur kernel, noise levels) are the same as the default DPS settings in the implementation. Search algorithms use $N = 8$ particles and resampling base $B = 16$ for both RFJ and Greedy search. Gradient guidance is applied with scale 0.5.
- **DAPS:** Box inpainting with box size 96, super-resolution with downsampling rate $10\times$. Noise levels unchanged from original DAPS defaults. Search algorithms use $N = 8$ particles and resampling base $B = 4$, with gradient guidance scale 13.
- **MPGD:** Box inpainting with box size 64, super-resolution with downsampling rate $6\times$, and Gaussian deblur with intensity of 5.0. Search algorithms use $N = 8$ particles and resampling base $B = 8$. Gradient guidance scale is 0.5 for box inpainting and 0.25 for super-resolution and Gaussian deblur.

C.2 Text Side Information Experiments

When text descriptions were used as side information, we made the degradation more severe so that the information in $\mathbf{s}$ was not already present in the measurement $\mathbf{y}$. Otherwise, side information would not provide meaningful guidance. For example, if the input image is sharp enough to identify the type of animal, then explicitly stating it in $\mathbf{s}$ adds little value.

The settings for these tasks are:

- **Box inpainting:** Box size 138, noise level same as default.
- **Super-resolution:** Downsampling rate $32\times$, noise level same as default.
- **Motion/Gaussian deblur:** Kernel size 256, intensity of Gaussian 5.0, noise level 0.1.
- **Nonlinear/Blind deblur:** Kernel size unchanged, noise level 0.5.

For all tasks in this setting, search algorithms use $N = 4$ particles and resampling base $B = 100$.

These hyperparameters ensure that our framework is evaluated under severe degradations (heavy downsampling, blurring, or noise), while search and guidance settings remain consistent across samplers and modalities.

C.3 MRI Experiments

We used the contrast-pairings among the files in the fastMRI dataset, provided by Atalık et al. (2025). We collect the data from the (fastMRI) source and preprocess to be compatible with the inputs in ContextMRI. Specifically, the setup used in the data is multi-coil MRI acquisition, which requires us to estimate the coil sensitivity maps, and then a complex reconstruction from them. ContextMRI

takes complex values as inputs and denoises to produce a complex-valued 2D image. We computed NMI with 64 bins at each step of the diffusion process to balance complexity with performance. We use the defaults parameters as in ContextMRI, except for the acceleration factor, 16 and the center fraction (ACS), 0.02. We use a pair of anatomy which two contrasts, which has more 30 slices. We consider the slices 15 to 28 as these are more challenging and report the results by using one as the side information for the other.

D LIMITATIONS

Our proposed search algorithms lack formal optimality guarantees for exploration–exploitation, and we do not claim they are theoretically optimal. We expect that stronger algorithms are possible, potentially improving both sample efficiency and robustness. A central reason is the absence of a general mathematical framework for designing optimal exploration–exploitation strategies in diffusion-based inverse problems with side information, an open problem we highlight. Practically, this means our methods rely on principled heuristics and tuned schedules (e.g., reward scaling, resampling/branching rates) whose compute allocation is not provably optimal, suggesting a clear direction for future work.

E USE OF LARGE LANGUAGE MODELS

Parts of this work were assisted by large language models (specifically GPT-5 from OpenAI). Their use was limited to improving clarity, grammar, and the presentation of experimental descriptions. All conceptual contributions, experimental design, analysis, and final decisions are solely the authors' responsibility. The models were not used to generate new research ideas, design experiments, or make unverifiable scientific claims. Additionally, large language models were used to generate textual descriptions serving as side information for a specific set of experiments, as detailed in the main paper.