Inference-Time Search Using Side Information for Diffusion-Based Image Reconstruction

Mahdi Farahbakhsh[†] **Vishnu Teja Kunde**[*] **Dileep Kalathil**
Krishna Narayanan **Jean-Francois Chamberland**
Department of Electrical and Computer Engineering, Texas A&M University

Abstract

Diffusion models have been used as 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 framework that incorporates side information into existing diffusion-based inverse problem solvers via inference-time search, in a plug-and-play, training-free manner. Through extensive experiments across a range of inverse problems, including inpainting, super-resolution, and several deblurring tasks, and across multiple diffusion-based inverse problem solvers (DPS, DAPS, and MPGD), we show that augmenting each solver with our framework consistently improves the quality of the reconstructions over the corresponding original method. To demonstrate the generality of our approach, we consider diverse forms of side information, including reference images, textual descriptions, and anatomical MRI scans. Code is available at this repository.[3]

1 Introduction

Diffusion models [Ho et al., 2020, Song et al., 2021b] have shown remarkable success in diverse generative tasks like text-to-image synthesis [Rombach et al., 2022], protein generation [Wu et al., 2024], video [Ho et al., 2022], audio [Kong et al., 2021], and language modeling [Sahoo et al., 2024]. Beyond generation, these models have shown great promise in **solving inverse problems**, where the goal is to reconstruct an image from partial or noisy measurements [Chung et al., 2023b, Daras et al., 2024]. A key distinction is that unlike many generative tasks (e.g., text-to-image synthesis, personalized editing, style transfer), which are often judged subjectively, inverse problems have a precise objective: to recover the ground-truth signal from incomplete measurements.

When the measurement is heavily degraded, the inverse problem becomes highly ill-posed as many reconstructions can explain the measurements almost equally well, and the measurement alone cannot tell them apart. A practical solution is to incorporate **side information**, defined as auxiliary measurements perceptually related to the target signal, to constrain the solution space and guide reconstruction. This idea is well established in classical signal processing, where 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, complementary measurements or modalities, such as multiple MRI contrasts, multimodal microscopy, or RGB guidance for NIR imaging, have been shown to substantially improve quality [Atalık et al., 2025, Tsiligianni and Deligiannis, 2019].

While the existing works on diffusion-based solvers have made significant progress on 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

[*]Equal Contribution.

[†]{mahdi.farahbakhsh,vishnukunde,dileep.kalathil,krn,chmbrlnd}@tamu.edu

[3]https://github.com/mhdfb/sideinfo-search-reconstruction

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

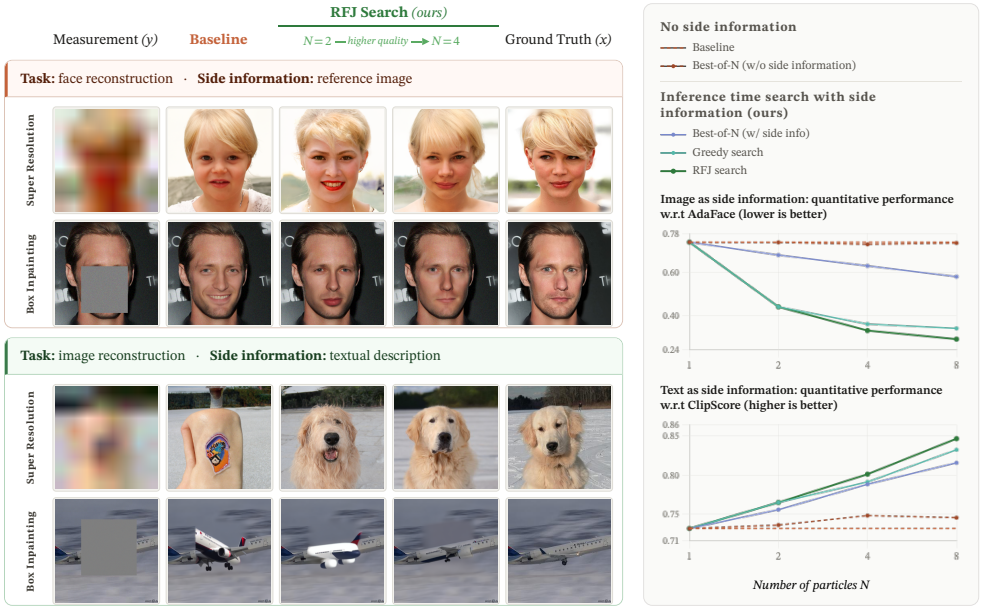


Figure 1: Incorporating side information via inference-time search improves reconstructions. (Left) Our framework augments state-of-the-art inverse problem solvers [Chung et al., 2023b,a, Zhang et al., 2025, He et al., 2024] with side information through inference-time search, yielding perceptually superior reconstructions with quality further improving as the number of particles N increases. (Right) All search algorithms within our side-information framework scale significantly better with additional compute than baselines without side information, demonstrating our framework's effectiveness.

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 model to take a specific side information 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 an approach that incorporates arbitrary side information via an auxiliary reward. This abstraction decouples the measurement model from the side information, is modality-agnostic (text, image, features), and requires no retraining. We derive tractable approximations with error bounds for the conditional score functions needed for sampling from pre-trained diffusion models.
- **Algorithm:** Motivated by inference-time search successes in LLMs [Snell et al., 2025, Setlur et al., 2025, Liu et al., 2025], we incorporate side information via inference time search. We use an off-the-shelf reward function scoring reconstructions by consistency with the side information. We consider two search strategies: Greedy Search (GS), an SMC-based approach that periodically selects the highest reward particles, and Recursive Fork-Join Search (RFJS), which resamples within dynamic particle groups, preserving diversity and balancing exploration and exploitation. Our approach plugs into standard inverse problem solvers and supports black-box, non-differentiable reward functions. We demonstrate that inference-time search with side information yields substantial gains even with moderate compute (N=4) (Fig. 1, left), whereas without side information, scaling provides negligible improvement, as seen in Best-of-N without side information (Fig. 1, right).
- **Experiments:** We evaluate across linear and nonlinear inverse problems (e.g., box inpainting, super-resolution, and motion/Gaussian/nonlinear deblurring) and multiple side information types (reference images, textual descriptions, and MRI). Applying our method to several diffusion-based inverse problem solvers (DPS [Chung et al., 2023b], Blind DPS [Chung et al., 2023a], DAPS

[Zhang et al., 2025], and MPGD [He et al., 2024]) we show consistent improvements in perceptual reconstruction quality over all baselines.

- **Ablation:** We provide extensive ablation studies demonstrating robustness to the choice of reward function (Appendix C.11), side information quality (Appendix C.7.2), and the resampling hyperparameter B across a range of values (Appendix C.12). They also confirm that scaling inference-time compute *without* side information fails to improve, or can even degrade, reconstruction quality (Table 11), justifying the additional compute cost.

2 Related work

Inverse problems with diffusion priors: Diffusion models [Dhariwal and Nichol, 2021, Ho et al., 2020, Song and Ermon, 2019, Sohl-Dickstein et al., 2015, Song and 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. Recent work has focused on improving performance by leveraging the manifold constraints [He et al., 2024] and by decoupling diffusion steps [Zhang et al., 2025]. 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. *None of these methods, however, 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. 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]. In MRI, LeSITA [Tsiligianni and Deligiannis, 2019] learns coupled sparse representations, and TGVN [Atalık et al., 2025] constrains ambiguous subspaces with additional contrasts using learned unrolled networks. Most approaches, however, are training-based or bound to one modality of side information associated with the trained conditional diffusion model.

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.

SMC methods: Sequential Monte Carlo approaches [Cardoso et al., 2024, Dou and Song, 2024, Wu et al., 2023, Achituve et al., 2025] generate and resample particles according to how well they fit the measurement. These methods, however, do not leverage side information.

Other related work: A detailed additional related work is provided in Section A.

3 Preliminaries and Problem Formulation

3.1 Preliminaries

Diffusion models: Diffusion models [Ho et al., 2020, Song et al., 2021b] are powerful generative models in which a neural network learns to reverse a forward process that progressively corrupts samples from p_{data} with Gaussian noise, enabling generation by iterative denoising. 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 $\mathbf{w}_t$ 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 non-negative monotonically increasing $\beta(\cdot)$. The corresponding reverse process of this SDE is described by [Anderson, 1982, Song et al., 2021b] $\mathrm{d}\mathbf{x}_t = \big(f(\mathbf{x}_t, t) - g^2(t)\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\big)\,\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}_t,\ \forall t \in [T, 0]$, where p_t is the marginal of $\mathbf{x}_t$, $\mathbf{x}_T \sim \mathcal{N}(0, I)$, and $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ represents the *score function*. Since p_t is unknown, the score function is approximated by a neural network $\mathcal{D}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ via score-matching. In practice, the SDE is discretized into T steps with $\alpha_t \triangleq \prod_{s=1}^{t}(1 - \beta_s)$.

Solving inverse problems using diffusion models: An inverse problem consists of recovering an unknown signal $\mathbf{x}_0$ from noisy, partial measurements $\mathbf{y} = \mathbf{A}(\mathbf{x}_0) + \sigma_y \mathbf{z}$, where $\mathbf{A}$ is the measurement

model, σ_y is the measurement noise level, and $\mathbf{z}$ is typically a Gaussian noise. Often, $\mathbf{A}$ is non-injective, making the problem ill-posed. A standard approach is Bayesian: assume a prior p_0 over the signal $\mathbf{x}_0$ and sample from the posterior $\mathbf{x}_0 \sim p_{0|Y}(\cdot \mid \mathbf{y})$. Though $p_{0|Y}$ is unknown, sampling can be achieved by replacing the score function in the backward SDE 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 addressed 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 $\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t) = (1/\sqrt{\alpha_t})(\mathbf{x}_t + (1-\alpha_t)\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)) \approx (1/\sqrt{\alpha_t})(\mathbf{x}_t + (1-\alpha_t)\mathcal{D}_{\boldsymbol{\theta}}(\mathbf{x}_t, t))$.

3.2 Problem formulation: solving inverse problems with side information

In many applications, the measurement $\mathbf{y}$ 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. Throughout, $\mathbf{s}$ is acquired independently of the measurement rather than computed from it, since anything derived from $\mathbf{y}$ could add nothing that $\mathbf{y}$ does not already contain. 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 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 $\mathbf{s}$; 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, complicating conditional-score construction. Second, even for measurement-only guidance, DPS-style algorithms require 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, which has shown remarkable gains in LLMs [Setlur et al., 2025, Liu et al., 2025, Snell et al., 2025] and text-conditioned diffusion [Singhal et al., 2025, Kim et al., 2025b], but not yet in 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 candidate 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: it is typically non-Gaussian, multi-modal, and modality dependent. 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 identity similarity. Such rewards are typically available and serve as practical surrogates for $p_{0|S}$ without requiring an explicit conditional density model. Although $\mathbf{s}$ is written as a single signal, the formulation generalizes to K reference signals $\mathbf{s}^1, \ldots, \mathbf{s}^K$ by replacing $r(\mathbf{x}_0; \mathbf{s})$ with an aggregate such as $\frac{1}{K}\sum_{k=1}^{K} r(\mathbf{x}_0; \mathbf{s}^k)$, which rewards equal alignment to all references, or $\max_{1\le k\le K} r(\mathbf{x}_0; \mathbf{s}^k)$, which rewards alignment to the closest reference (see Table 15 for a quantitative comparison).

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 LLM alignment [Ouyang et al., 2022, Rafailov et al., 2023], where generation is steered toward high-reward samples without deviating too much from the pre-trained prior 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, we *define* our tractable surrogate for $p_{0|S}$ as

$$p_{0|S}(\mathbf{x}_0 \mid \mathbf{s}) \propto p_0(\mathbf{x}_0) \exp\left(\tfrac{r(\mathbf{x}_0;\mathbf{s})}{\tau}\right), \tag{1}$$

Comparing Eq. (1) with Bayes' rule, $p_{0|S}(\mathbf{x}_0 \mid \mathbf{s}) = p_{S|0}(\mathbf{s} \mid \mathbf{x}_0)p_0(\mathbf{x}_0)/p_S(\mathbf{s})$, the above surrogate becomes the true posterior precisely when, for every $\mathbf{x}_0$

$$\frac{r(\mathbf{x}_0; \mathbf{s})}{\tau} = \log p_{S|0}(\mathbf{s} \mid \mathbf{x}_0) + c(\mathbf{s}) \tag{2}$$

for some c that does not depend on $\mathbf{x}_0$; that is, when the reward is an affine function of the log-likelihood with slope $1/\tau$. For each reward we use, a capable pretrained network models $\log p_{S|0}(\mathbf{s} \mid \mathbf{x}_0)$ approximately and makes it computable in closed form from the network's own training objective, so Eq. (2) can be evaluated directly. Section B.2 derives this and Section C.16 measures it to show that the rewards used here meet the condition to a good approximation, which justifies Eq. (1) as a model of $p_{0|S}$. 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. (1); 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. (1) 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 Eq.* (1) *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 the value function is given by $V_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y}) \triangleq \log \mathbb{E}_{\mathbf{x}_0\sim p_{0|t,Y}(\cdot|\mathbf{x}_t,\mathbf{y})}[\exp(r(\mathbf{x}_0; \mathbf{s})/\tau)]$.

The proof is provided in Appendix B.1. Using Eq. (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

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

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

Appendix B.3 details the steps leading to Eq. (6)–Eq. (7), and provides a bound on the approximation error $|V_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y}) - \hat{V}_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y})|$ in Proposition 3.

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 (Eqs. (6)–(7)). 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 $\nabla_{\mathbf{x}_t} V_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y}) \approx \nabla_{\mathbf{x}_t} r(\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t); \mathbf{s})/\tau$. We show that the approximation error remains small when t is small even when $\eta = 0$, see Appendix B.3 and Section B.3.1. 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 calculating a gradient through the denoiser network at each step of the backward diffusion is computationally intensive. Moreover, RGG is highly sensitive to the hyperparameter, leading to limited performance improvements and undesirable artifacts in the reconstructed images, as illustrated in Section C.8. This motivates a gradient-free approach for leveraging the side information in inverse problems.

4.2 Inference-Time Search Algorithms for Inverse Problems

Inference-time search have recently gained popularity in LLMs [Snell et al., 2025, Setlur et al., 2025, Liu et al., 2025]. Search algorithms solve a multi-step decision-making problem, balancing exploration and exploitation. While Monte Carlo Tree Search (MCTS) [Kocsis and 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 $\mathbf{x}_t$ requires repeated rollouts. Using a value function to guide the search is an alternative, but this requires significant additional training for each modality of side information. These limitations motivate training-free, computationally tractable inference-time methods.

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 $N \to \infty$ [Wu et al., 2023, Dou and Song, 2024]. In practice, frequent resampling collapses diversity by repeatedly duplicating high-reward particles, leading to under-exploration of the solution space. This motivates a more balanced resampling strategy.

To address this, we introduce **grouped resampling**: we partition the N particles into groups of size g_t and resample independently within each group, so particles only compete locally. This preserves between-group diversity while still exploiting high-reward particles within each group. Based on how we form these groups, we introduce two search algorithms.

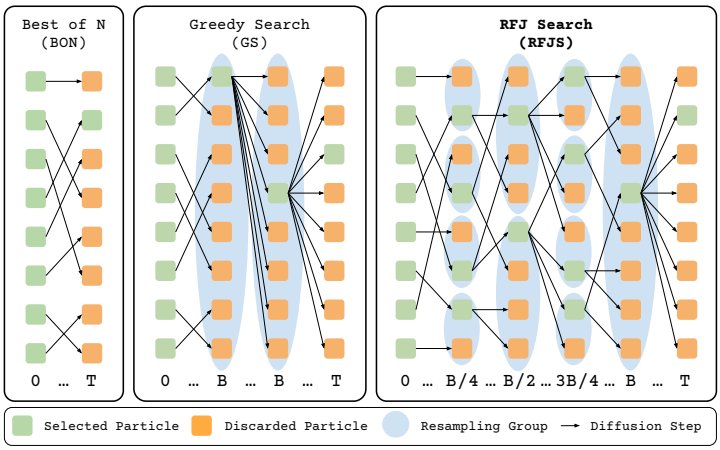
Greedy Search (GS): Here, we use a fixed resampling period B and select $g_t = N$ whenever $t \bmod B = 0$, and $g_t = 1$ otherwise. Greedy Search reduces to the **Best-of-N** (BON) strategy when $B \geq T$, since in that case $g_t = 1$ for all t. Smaller values of B emphasize short-term reward exploitation, while larger values promote exploration. An illustration of Greedy Search, with resampling interval B, is shown 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): To balance exploration and exploitation, we propose a hierarchical resampling schedule inspired by how researchers collaborate. As a PhD student, you meet your immediate collaborators frequently, your research group weekly, and the entire AI community once a year at conferences, and when a larger meeting is scheduled, the smaller one is canceled. We apply the same principle to particles: you have N particles in total. Every B steps, you resample across the entire group. Every $B/2$ steps, you resample within groups of size $N/2$, and so on. Formally, the group size at step t is $g_t = N/2^{j^*}$, where $j^* = \min\{\, i \geq 0 : \; t \bmod (B/2^i) = 0 \,\}$.

Smaller groups are resampled more often while larger groups are resampled less frequently, resembling the higher frequency of small group meetings compared to large conferences. Particles within each group search the solution space locally, while keeping distinct groups independent from each other ensures a global search over the space, naturally balancing exploration and exploitation.

Figure 2 illustrates this strategy, where resampling frequency decreases as group size increases. We have summarized this inference-time search framework in Algorithm 1 in Appendix B.4.

Figure 2: Illustration of inference-time search methods. Unlike GS, RFJS preserves diversity via dynamic grouping and within-group resampling, thus balancing exploration and exploitation.



Our framework is modular: regardless of the choice of search algorithm, it can be used to incorporate side information into the inference process. While any search algorithm can be paired with our framework, we find that RFJS scales better in our setting. Furthermore, RFJS simplifies hyperparameter tuning: because it naturally balances exploration and exploitation, τ (used for assigning the weights $w[i]$ for particles) can be set to a sufficiently small value ($\tau \to 0$) such that selection becomes deterministic, always retaining the particle with the highest reward, without sacrificing performance. Every experiment in this paper uses this setting, in which only the relative *ordering* of the rewards among the candidates matters and not their absolute values. For comparison, Table 21 reports the sampler at the positive temperatures $\tau = 1/10$ and $\tau = 1/3$, where the side-information metric improves as τ decreases, justifying deterministic selection as our default.

5 Experiments

5.1 Experimental Setup

We evaluate our inference-time search framework for solving inverse problems with side information by instantiating two search algorithms we proposed: **Greedy Search (GS)** and **Recursive Fork Join Search (RFJS)**. We consider two types of side information: (i) **image side information**, a reference image of the same entity (e.g., the same person under different poses or lighting), and (ii) **text side information**, a textual description of the target image.

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) **DAPS** [Zhang et al., 2025], and (iv) **MPGD** [He et al., 2024] (reported in Appendix C).

Tasks: We consider six inverse problems, covering both linear: (i) box inpainting, (ii) super resolution, (iii) motion deblur, (iv) Gaussian deblur, and nonlinear: (v) nonlinear deblur, (vi) blind deblur. See Appendix D for details.

Table 1: **Image as side information:** Quantitative results using DPS as the baseline solver. Our algorithms significantly improve perceptual quality as measured by FS, which is more aligned with human perception, while keeping classical metrics (PSNR, SSIM, LPIPS) in a similar range and slightly improving them.

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

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

Figure 3: **Image as side information:** Qualitative results using DPS and BlindDPS baseline solvers. In severely ill-posed inverse problems, baseline solvers can fail in identity preservation, while RFJS remains effective; in milder settings, RFJS recovers finer details.



Figure 4: **Image as side information:** Qualitative results with DAPS as baseline solver.

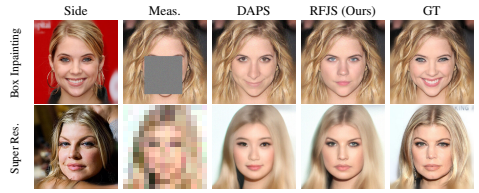


Table 2: **Image as side information:** Quantitative results with DAPS as the baseline solver.

	Box Inpainting				Super Resolution (×10)			
Algorithm	FS (↓)	PSNR (↑)	LPIPS (↓)	SSIM (↑)	FS (↓)	PSNR (↑)	LPIPS (↓)	SSIM (↑)
RFJS (ours)	**0.423**	**28.720**	**0.140**	**0.788**	0.654	25.228	**0.282**	0.661
GS (ours)	0.511	28.640	0.140	0.787	0.760	**25.271**	0.285	**0.662**
RGG	0.436	28.410	0.141	0.784	**0.579**	25.210	0.282	0.659
BON (w/ side)	0.611	28.660	0.141	0.787	0.909	25.220	0.285	0.660
DAPS	0.739	28.290	0.142	0.784	1.020	25.170	0.285	0.659

the approximation $\nabla_{\mathbf{x}_t} V_t^\tau(\mathbf{x}_t; \mathbf{s}, \mathbf{y}) \approx \nabla_{\mathbf{x}_t} r(\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t); \mathbf{s})/\tau$. Experiments are run on NVIDIA A100 GPUs.

5.2 Main Results

Image as side information: The goal is to reconstruct a face image from a noisy measurement when another image of the same identity is available (Fig. 3, Fig. 4, Table 1, Table 2). Using CelebA-HQ [Karras et al., 2018] 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 FS, computed as the distance between the identity embeddings of the reconstructed and side-information faces, extracted by pretrained AdaFace network.

Since standard metrics (PSNR, SSIM, LPIPS) often fail to capture identity similarity, we additionally use the face-similarity distance FS (↓), the distance between the AdaFace identity embeddings of two faces, so that smaller values mean greater similarity. Reported as a metric, FS compares the reconstruction with the ground truth; the search reward is the negative of the same distance taken against the side information instead. We use the name FS throughout. Table 1 and Table 2 show that both proposed inference-time search methods, GS and RFJS, outperform baselines (DPS and DAPS respectively), with RFJS achieving the best overall scores. Qualitative results in Fig 3 show sharper facial details and preserve identity. We used $N = 8$ particles. The resampling base B is fixed once for the base sampler and then used unchanged for every task, with no per-task tuning: in the face experiments it is $B = 16$ with DPS, 4 with DAPS and 8 with MPGD. Section C.12 reports the sensitivity to this choice, and Section D lists the value used in each experiment.

Text as side information: The goal is to reconstruct an image from its noisy measurement, with a text description of the image available as side information. We use a pre-trained diffusion model trained on the ImageNet data [Dhariwal and 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 as the side information. 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 $32\times$ 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

Figure 5: **Text as side information:** Qualitative results using textual side information under severe degradation (e.g., 32× super-resolution with the prompt *"golden retriever sitting on a snowy frozen lake, facing forward"*). RFJS recovers perceptually meaningful content while DPS fails.

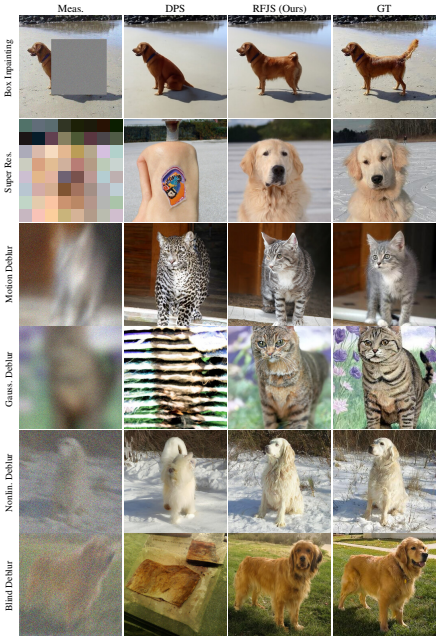
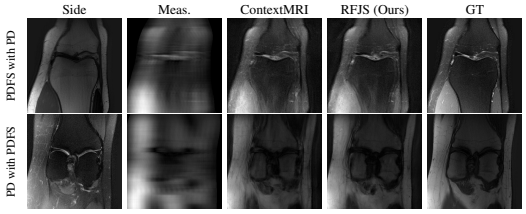


Table 3: **Text as side information:** Quantitative comparison of our algorithms with the baseline. RFJS and GS attain the best CS on every task and the best value in 19 of the 24 cells, with Best-of-N leading in the remaining five.

	Box Inpainting				Super Resolution (×32)			
Algorithm	CS (↑)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CS (↑)	PSNR (↑)	SSIM (↑)	LPIPS (↓)
RFJS (ours)	**0.901**	**20.75**	**0.678**	**0.294**	**0.801**	17.13	**0.352**	**0.493**
GS (ours)	0.894	19.76	0.676	0.305	0.791	17.20	0.351	0.509
BON (w/ side)	0.882	19.99	0.672	0.308	0.788	**17.21**	0.350	0.500
DPS	0.871	19.86	0.672	0.312	0.731	16.90	0.330	0.522
	Non-linear Deblur				Motion Deblur			
Algorithm	CS (↑)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CS (↑)	PSNR (↑)	SSIM (↑)	LPIPS (↓)
RFJS (ours)	0.863	**20.58**	**0.473**	**0.405**	**0.858**	18.61	0.402	**0.424**
GS (ours)	**0.865**	20.32	0.456	0.405	0.835	17.83	0.369	0.453
BON (w/ side)	0.855	20.52	0.464	0.406	0.848	**19.24**	**0.415**	0.427
DPS	0.839	20.55	0.469	0.409	0.794	18.16	0.384	0.458
	Gaussian Deblur				Blind Deblur			
Algorithm	CS (↑)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CS (↑)	PSNR (↑)	SSIM (↑)	LPIPS (↓)
RFJS (ours)	**0.843**	**18.10**	0.358	0.457	**0.851**	18.84	0.412	**0.433**
GS (ours)	0.835	17.96	0.356	0.457	0.835	**18.93**	**0.414**	0.438
BON (w/ side)	0.831	17.99	**0.365**	**0.452**	0.831	18.78	0.410	0.443
DPS	0.778	16.79	0.329	0.487	0.793	18.82	0.409	0.459

Figure 6: **Contrast Image as Side Information:** Qualitative MRI reconstruction with RFJS vs. ContextMRI. The shapes and line edges are well preserved in our reconstruction.



and textual alignment. The qualitative reconstructions in Fig. 5 closely match the textual descriptions, and the quantitative metrics in Table 3 show that both GS and RFJS outperform competing baselines. Section C.3 repeats this evaluation on 75 images.

MRI with multi-contrast side information: We include this modality to show that the framework carries over to scientific imaging, where the side information is a second acquisition of the same anatomy rather than a photograph or a caption (Section D.3 gives the setup in full). We use 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 16× undersampling with 2% ACS. We use normalized mutual information (NMI) as reward, which is robust to contrast changes. Fig. 6 highlights sharper edges and more faithful structure. Table 26 in Appendix D.3 shows our methods consistently improve the baseline.

Role of side information in enabling inference-time scaling: A central contribution of our framework is the principled use of side information through a reward that measures consistency between the reconstruction and the side information at inference time. To see why this is essential, consider the natural alternative: scaling compute via Best-of-N using only the measurement residual $\|\mathbf{y} - \mathbf{A}\mathbf{x}_0^i\|$ as the selection criterion (BoN w/o SI). When $\mathbf{y}$ is noisy or the inverse problem is ill-posed, this criterion is actively misleading as it favors particles that explain the noise rather than the underlying signal, and as Table 4 shows, performance *can degrade* relative to the baseline. Incorporating side information fundamentally changes this picture: BoN with our reward (BoN w/ SI) draws samples that are more faithful to the conditional posterior $p_{X|Y,S}$, resolving ambiguities that the measurement alone cannot (see Table 11 for more tasks).

Table 4: Without side information, Best-of-N selection by $\|\mathbf{y} - \mathbf{A}\mathbf{x}_0^i\|$ fails to improve over the baseline in noisy settings. With side information, selection by our reward $r(\mathbf{x}_0; \mathbf{s})$ consistently improves all metrics.

	Box Inpainting				Super Resolution			
Algorithm	PSNR (↑)	LPIPS (↓)	SSIM (↑)	CS (↑)	PSNR (↑)	LPIPS (↓)	SSIM (↑)	CS (↑)
DAPS	18.21	0.274	0.742	0.855	18.44	0.583	0.391	0.727
BON (w/o SI)	18.02	0.274	0.738	0.846	18.13	0.590	0.368	0.729
BON (w/ SI, ours)	**19.51**	**0.262**	**0.750**	**0.893**	**18.65**	**0.579**	**0.398**	**0.767**

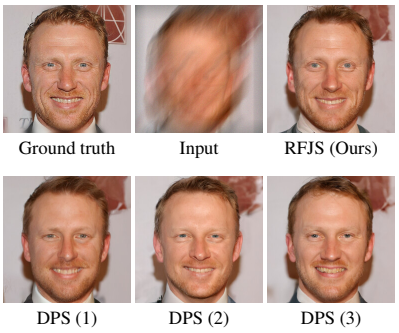
	Box Inpainting		Super Resolution $\times 16$		Motion Deblur $K{=}256$	
Metric	DPS	RFJS (ours)	DPS	RFJS (ours)	DPS	RFJS (ours)
DreamSim (↓)	0.078	**0.054** (+30.8%)	0.178	**0.127** (+29.0%)	0.120	**0.088** (+27.2%)
CLIP (↓)	0.088	**0.061** (+30.1%)	0.202	**0.136** (+32.7%)	0.123	**0.085** (+30.4%)
NME (↓)	0.0275	**0.0248** (+9.7%)	0.0331	**0.0318** (+4.1%)	0.0292	**0.0271** (+7.0%)
FS (↓)	0.740	**0.298** (+59.8%)	1.310	**0.481** (+63.3%)	1.212	**0.439** (+63.8%)
PSNR (↑)	27.21	**27.78**	20.98	**21.04**	22.18	**22.20**
LPIPS (↓)	0.141	**0.137**	0.296	**0.293**	0.263	**0.261**
SSIM (↑)	0.846	**0.852**	0.563	**0.567**	0.604	**0.607**

Table 5: Search with side information improves other perceptual metrics which are not directly derived from the reward modeling network. The last two tasks use the severe-degradation setting. See Table 6 for the full set of tasks.

Perceptual Quality vs. Classical Metrics: Recent works [Lin et al., 2025, Su et al., 2025] have shown that improvements in perceptual quality for inverse problems do not necessarily translate into better values of classical metrics such as PSNR, SSIM, and LPIPS, and they advocate evaluating with task-specific metrics. In our setting, we use FS for face reconstruction and CLIPScore for ImageNet as task-specific metrics, which are also more aligned with human perceptual judgments for these tasks. Because FS and the AdaFace reward share a model, we additionally report three other perceptual metrics not derived from the same model: DreamSim [Fu et al., 2023], a perceptual distance fit to human similarity judgements; the CLIP distance between the reconstruction and the ground truth; and the normalised landmark error NME, the mean distance between the 68 FAN landmarks [Bulat and Tzimiropoulos, 2017] of the reconstruction and of the ground truth over the latter's inter-ocular distance, which measures facial geometry rather than identity. All three improve alongside FS (Table 5, and all twelve tasks in Table 6) indicating better (perceptual) reconstruction rather than simply target reward optimization.

Fig. 7 provides a clear qualitative example of this phenomenon. Although our RFJS reconstruction is more faithful to the ground truth in terms of identity preservation, all three BlindDPS reconstructions achieve better PSNR, SSIM, and LPIPS values. This illustrates that classical metrics can fail to capture meaningful semantic improvements, even when perceptual quality is clearly superior. We provide additional examples and quantitative evidence in Appendix C.1, demonstrating large perceptual gains are not reflected in PSNR, SSIM, or LPIPS.

Figure 7: Classical metrics fail to capture perceptual fidelity: three BlindDPS reconstructions achieve higher PSNR values (23.5, 23.6, 23.9) than RFJS (23.2), yet RFJS exhibits clearly stronger perceptual similarity to the ground truth. The same pattern holds for LPIPS/SSIM.



5.3 Additional Experiments (Appendix)

The results of our framework for other samplers like MPGD, along with additional DPS results, are provided in Appendix C. We further include ablation/sensitivity analyses that address **compute, runtime, robustness, and sensitivity**: (i) **Compute–performance trade-off:** effect of the **number of particles** (Section C.9). (ii) **Runtime:** wall-clock comparison across methods/settings (Section C.10). (iii) **Robustness:** degradation of **measurement quality** (harder inverse problems) (Section C.7.1) and **side-information quality** (Section C.7.2). (iv) **Sensitivity of Gradient Guidance:** motivating our search-based alternative (Section C.8). (v) **Toy examples:** illustrating the benefits of side information (Section C.13).

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.

Acknowledgments and Disclosure of Funding

We thank the department of Electrical and Computer Engineering (ECEN) at Texas A&M University for providing access to the Olympus computing cluster. Portions of this research were conducted with the advanced computing resources provided by Texas A&M High Performance Research Computing. This work was supported, in part, by the U.S. Army Combat Capabilities Development Command (DEVCOM) under Grant Number W911NF2520046. This work was also supported in part by the National Science Foundation grant NSF-CNS 2148354, federal agencies, and industry partners as specified in the Resilient & Intelligent NextG Systems (RINGS) program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

The authors declare no competing interests in relation to this work.

References

Idan Achituve, Hai Victor Habi, Amir Rosenfeld, Arnon Netzer, Idit Diamant, and Ethan Fetaya. Inverse Problem Sampling in Latent Space Using Sequential Monte Carlo. In *International Conference on Machine Learning (ICML)*, 2025.

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.

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

Ralph Allan Bradley and Milton E. Terry. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1021–1030, 2017.

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

Ben Fei, Zhaoyang Lyu, Liang Pan, Junzhe Zhang, Weidong Yang, Tianyue Luo, Bo Zhang, and Bo Dai. Generative diffusion prior for unified image restoration and enhancement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023.

Stephanie Fu, Netanel Yakir Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. DreamSim: Learning new dimensions of human visual similarity using synthetic data. In *Advances in Neural Information Processing Systems*, volume 36, pages 50742–50768, 2023.

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

Chi-Wei Hsiao, Yu-Lun Liu, Cheng-Kun Yang, Sheng-Po Kuo, Yucheun Jou, and Chia-Ping Chen. ReF-LDM: A Latent Diffusion Model for Reference-based Face Image Restoration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Michael F. Hutchinson. A Stochastic Estimator of the Trace of the Influence Matrix for Laplacian Smoothing Splines. *Communications in Statistics – Simulation and Computation*, 18(3):1059–1076, 1989.

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.

Yazid Janati, Badr MOUFAD, Alain Oliviero Durmus, Eric Moulines, and Jimmy Olsson. Divide-and-conquer posterior sampling for denoising diffusion priors. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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.

Aayush Karan, Kulin Shah, and Sitan Chen. ReGuidance: A Simple Diffusion Wrapper for Boosting Sample Quality on Hard Inverse Problems, 2025. URL https://arxiv.org/abs/2506.10955.

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

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 (ICLR)*, 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 *International Conference on Learning Representations (ICLR)*, 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 (ICLR)*, 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.

Tianyu Lin, Xinran Li, Chuntung Zhuang, Qi Chen, Yuanhao Cai, Kai Ding, Alan L. Yuille, and Zongwei Zhou. Are pixel-wise metrics reliable for sparse-view computed tomography reconstruction?, 2025. URL https://arxiv.org/abs/2506.02093.

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.

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.

Kushagra Pandey, Farrin Marouf Sofian, Felix Draxler, Theofanis Karaletsos, and Stephan Mandt. Variational control for guidance in diffusion models. In *International Conference on Machine Learning (ICML)*, 2025.

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.

Minsi Ren, Wenhao Deng, Ruiqi Feng, and Tailin Wu. One Step Further with Monte-Carlo Sampler to Guide Diffusion Better. In *International Conference on Learning Representations (ICLR)*, 2026.

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 *International Conference on Learning Representations (ICLR)*, 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 *International Conference on Learning Representations (ICLR)*, 2025.

David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 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. In *International Conference on Machine Learning*, 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 *International Conference on Learning Representations (ICLR)*, 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 *International Conference on Learning Representations (ICLR)*, 2024.

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

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*, 2023.

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 (ICLR)*, 2021b.

Shaolin Su, Josep M. Rocafort, Danna Xue, David Serrano-Lozano, Lei Sun, and Javier Vazquez-Corral. Rethinking image evaluation in super-resolution, 2025. URL https://arxiv.org/abs/2503.13074.

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.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

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

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)*, 2024.

Appendices

A Additional Related Work

Reward-gradient guidance: LGD [Song et al., 2023] 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. More recently, ABMS [Ren et al., 2026] reduces the estimation error of this gradient by inserting an additional backward denoising step and averaging over M Monte Carlo samples of $\mathbf{x}_0$ before differentiating. It maintains no particle weights and performs no resampling, and is therefore best understood as a lower-bias variant of the RGG baseline we compare against (Section 5) rather than as a particle-based method. In addition to being gradient-based approaches, they are typically used for semantic generation tasks rather than inverse problems, and they all require the reward to be differentiable.

SMC methods: Sequential Monte Carlo approaches [Cardoso et al., 2024, Dou and Song, 2024, Wu et al., 2023] generate and resample particles under tilted distributions, offering gradient-free alternatives. 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. LD-SMC [Achituve et al., 2025] extends this construction to the latent space of a diffusion model.

An SMC method is specified by two choices: how particles are weighted, and how they are resampled. The methods above weight particles by the measurement $\mathbf{y}$ and resample over the full particle set. Our algorithms differ in both respects. The weights come from the side information, through the reward $r(\mathbf{x}_0; \mathbf{s})$, which carries information that $\mathbf{y}$ does not when the inverse problem is severely ill-posed. Resampling is local: particles in RFJS are resampled within dynamically formed groups rather than globally. Greedy Search is the special case of a single global group, and is in this sense an SMC algorithm with side-information weights, while RFJS generalizes it along the resampling axis. Section C.14 reports comparisons with other weighting and resampling rules, namely adaptive and annealed SMC, stratified and multinomial resampling, adaptive group sizes and reward-guided beam search, together with stochastic resampling at a positive temperature. Proposition 1 holds for every $\tau > 0$, and the deterministic selection used in our main experiments is its $\tau \to 0$ limit.

Reference-based restoration with trained conditional models: A complementary line of work trains the generative model itself to consume the side information. ReF-LDM [Hsiao et al., 2024] is a latent diffusion model for reference-based face restoration: it conditions on one or several reference images of the same identity through a cached key–value mechanism and is trained with a timestep-scaled identity loss on a paired dataset of degraded images and references. Because the references are available during training, such a model can aggregate several of them jointly, and restoration quality improves as more references are supplied [Hsiao et al., 2024]. What this requires is a paired dataset of targets and side information, a training run, and a fixed choice of degradation process and side-information modality. Section C.15 compares the two approaches on super-resolution and on motion deblurring. Multiple references enter our framework at the level of the reward, by aggregating per-reference rewards (Section C.7.2).

Text-conditioned solvers and training-free wrappers: TReg [Kim et al., 2025a] uses a textual description as side information by regularizing the reverse diffusion of a text-conditioned latent model through that model's own conditioning interface. Since our search runs on top of an arbitrary sampler, TReg can also serve as a base solver for it, and we report this combination in Section C.15. ReGuidance [Karan et al., 2025] is a training-free wrapper that acts on the initialization rather than the guidance: it inverts a candidate reconstruction by running the unconditional probability-flow ODE in reverse and uses the resulting latent to initialize DPS, which improves realism and measurement consistency on hard inverse problems. Like our framework it reuses an existing solver as a black box, but it does not use side information.

Guidance and side information in diffusion models: Several recent works have explored conditioning, guidance, and optimization strategies for diffusion models, but they differ substantially in whether and how *side information* is incorporated.

CLAY [Zhang et al., 2024] is a conditional generative model for 3D asset synthesis that incorporates text, image views, and simple 3D cues via trained, modality-specific conditioning modules (e.g., cross-attention adapters) within a native 3D diffusion model. While this enables controllable generation,

the conditioning signals are integrated through learned modules during training, and the method is not formulated as an inverse problem under a measurement model.

The term side information is also used by Fei et al. [2023], but in a different sense: their setting considers multiple measurements of the same underlying ground truth, where the relationship between measurements and the clean image is either known or can be estimated parametrically. As a result, the additional information can be incorporated through an explicit forward-model-based formulation. In contrast, we consider side information that may come from heterogeneous modalities (e.g., text, reference images, alternative MRI contrasts) with unknown and nonlinear relationships to the target image. Since such relationships are not expressible as forward models, we incorporate side information implicitly via a reward that guides inference-time search, without retraining.

Other inference-time optimization approaches for diffusion models do not incorporate side information. For example, Chung et al. [2024] optimizes latent variables and text-prompt embeddings during inference to better exploit the learned prior, improving alignment with the prompt but relying solely on internal model variables. Similarly, Janati et al. [2024] uses distribution tilting aligned with the measurement via Langevin dynamics, *without incorporating external side information*, and does not perform inference-time search.

Finally, Pandey et al. [2025] proposes a variational framework that optimizes a reward defined with respect to the measurement likelihood and assumes differentiable reward functions. By contrast, we model the relationship between the target image and side information using a reward-tilted posterior and support non-differentiable rewards, enabling flexible integration of arbitrary side information at inference time.

B Proofs

B.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

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

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

B.2 Exactness Conditions for the Reward-Tilted Target

Equation (2) records what would have to be true of the reward for the tilted distribution of Eq. (1) to be the conditional law of $\mathbf{x}_0$ given $\mathbf{s}$: the reward must be an affine function of $\log p_{S|0}(\mathbf{s} \mid \mathbf{x}_0)$ with slope $1/\tau$, up to a term that does not depend on $\mathbf{x}_0$. Each of the two constructions below pairs a reward used in this paper with a particular temperature rather than leaving τ free.

Both rewards used in this paper are obtained by training a network, and in each case the training objective determines, at its population optimum, a quantity from which $\log p_{S|0}(\mathbf{s} \mid \mathbf{x}_0)$ can be recovered *exactly*. The two statements below make this precise. Neither requires a model of how $\mathbf{s}$ was generated; both require only that the network has the capacity to attain its own optimum.

Face identity. Here $\mathbf{s}$ is a photograph of the same person as $\mathbf{x}_0$. Let $y \in \{1, \ldots, M\}$ be the identity, and let $\pi(\mathbf{x}) \in \Delta^{M-1}$ denote the classifier's softmax, $\pi_j(\mathbf{x}) = \mathrm{softmax}_j\big(\kappa\langle\Phi(\mathbf{x}), \mathbf{w}_j\rangle\big)$, which is what AdaFace [Kim et al., 2022] is trained to produce: a normalised softmax over identities with ℓ_2-normalised embeddings Φ and class weights $\mathbf{w}_j$, at logit scale κ. Assume

- **(A1)** *capacity*: the classifier attains the population minimum of its cross-entropy objective, so $\pi_j(\mathbf{x}) = \Pr(y = j \mid \mathbf{x})$ for almost every $\mathbf{x}$;
- **(A2)** *conditional independence*: $\mathbf{s} \perp \mathbf{x}_0 \mid y$, two photographs of one person being independent given the identity;
- **(A3)** *uniform identity prior*: $\Pr(y = j) = 1/M$.

Proposition. Under (A1)–(A3), for every $\mathbf{x}_0$ and $\mathbf{s}$,

$$\log \frac{p_{S|0}(\mathbf{s} \mid \mathbf{x}_0)}{p_S(\mathbf{s})} = \log M + \log \big\langle \pi(\mathbf{x}_0), \pi(\mathbf{s}) \big\rangle. \tag{8}$$

Proof. Marginalising the identity and using (A2), $p_{S|0}(\mathbf{s} \mid \mathbf{x}_0) = \sum_j p(\mathbf{s} \mid y = j) \Pr(y = j \mid \mathbf{x}_0)$. Bayes' rule gives $p(\mathbf{s} \mid y = j) = \Pr(y = j \mid \mathbf{s})\, p_S(\mathbf{s}) / \Pr(y = j)$, and (A3) makes the denominator $1/M$, so $p_{S|0}(\mathbf{s} \mid \mathbf{x}_0) = M\, p_S(\mathbf{s}) \sum_j \Pr(y = j \mid \mathbf{s}) \Pr(y = j \mid \mathbf{x}_0)$. By (A1) the two posteriors are $\pi(\mathbf{s})$ and $\pi(\mathbf{x}_0)$. Divide by $p_S(\mathbf{s})$ and take logarithms. □

The right-hand side is computable from the embeddings alone,

$$\big\langle \pi(\mathbf{x}_0), \pi(\mathbf{s}) \big\rangle = \sum_{j=1}^{M} \mathrm{softmax}_j\big(\kappa\langle\Phi(\mathbf{x}_0), \mathbf{w}_j\rangle\big)\, \mathrm{softmax}_j\big(\kappa\langle\Phi(\mathbf{s}), \mathbf{w}_j\rangle\big),$$

while $\log M$ and $\log p_S(\mathbf{s})$ do not depend on $\mathbf{x}_0$ and are therefore absorbed into $c(\mathbf{s})$. The quantity against which Eq. (2) measures a reward is thus available for any pair $(\mathbf{x}_0, \mathbf{s})$, which is what turns that condition from a statement about an unknown density into one that can be checked.

AdaFace is trained at $\kappa = 64$, a value suited to its own objective of deciding among the M identities: at that scale the softmax is close to one-hot, so $\langle \pi(\mathbf{x}_0), \pi(\mathbf{s}) \rangle$ is close to the indicator that $\mathbf{x}_0$ and $\mathbf{s}$ select the same identity. Our objective asks for the opposite. A reward must separate candidate reconstructions that already share an identity, which a binary verdict cannot do, so we evaluate the same expression at the smaller $\kappa = 8$. This keeps the form of Eq. (8) and yields the soft, graded tilt the search needs.

The reward that we use based on the embedding distance, $r = -\|\Phi(\mathbf{x}_0) - \Phi(\mathbf{s})\|_2$, varies continuously with $\langle \Phi(\mathbf{x}_0), \Phi(\mathbf{s}) \rangle$ and has the same soft distinction while being much simpler to interpret. Besides it has no scale to choose and no dependence on which M identities the network was trained to separate.

Textual descriptions. Here $\mathbf{s} = T$ is a description of $\mathbf{x}_0$ and the reward is the score $f(T, \mathbf{x}_0)$ of ImageReward [Xu et al., 2023], trained with the pairwise logistic (Bradley–Terry) loss [Bradley and Terry, 1952] $-\log \sigma\big(f(T, \mathbf{x}^1) - f(T, \mathbf{x}^2)\big)$ over annotator-preferred pairs $\mathbf{x}^1 \succ \mathbf{x}^2$. Assume

- **(B1)** *capacity*: f attains the population minimum of that loss;
- **(B2)** *annotators compare by likelihood*: $\Pr(\mathbf{x}^1 \succ \mathbf{x}^2 \mid T) = p_{S|0}(T \mid \mathbf{x}^1) / \big(p_{S|0}(T \mid \mathbf{x}^1) + p_{S|0}(T \mid \mathbf{x}^2)\big)$.

Proposition. Under (B1)–(B2) there is a function a of the prompt alone with

$$f(T, \mathbf{x}_0) = \log p_{S|0}(T \mid \mathbf{x}_0) + a(T), \tag{9}$$

so Eq. (2) holds exactly for $r = f$ at $\tau = 1$ and $c(\mathbf{s}) = a(T)$.

Proof. By (B1) the optimum satisfies $\sigma\big(f(T, \mathbf{x}^1) - f(T, \mathbf{x}^2)\big) = \Pr(\mathbf{x}^1 \succ \mathbf{x}^2 \mid T)$ for every pair, and by (B2) the right-hand side equals $\sigma\big(\log p_{S|0}(T \mid \mathbf{x}^1) - \log p_{S|0}(T \mid \mathbf{x}^2)\big)$. As σ is injective, $f(T, \cdot) - \log p_{S|0}(T \mid \cdot)$ takes the same value at $\mathbf{x}^1$ and $\mathbf{x}^2$ for every pair, hence is a function of T alone. □

Note that the logistic link has unit slope, so (B1) identifies the *scale* of f and not only its ordering; the single remaining degree of freedom is the additive $a(T)$, which Eq. (2) permits.

Both rewards used in this paper therefore satisfy Eq. (2) approximately. For textual side information it holds directly, since Eq. (9) makes the ImageReward score itself the required affine function at $\tau = 1$. For face identity the reward we run is the embedding distance rather than $\log\langle \pi(\mathbf{x}_0), \pi(\mathbf{s}) \rangle$, and Section C.16 settles the difference between the two by fitting one against the other, finding the embedding distance to be an approximately affine image of the log-likelihood ratio that Eq. (8) supplies.

Ordering in the small-τ limit. The experiments reported in this paper use the small-τ setting described in Section 4.2, in which resampling retains the highest-reward particle in each group. In that regime the weights $w[i] \propto \exp(r[i]/\tau)$ enter only through which of them is largest, so any strictly increasing transformation of r traces out the same trajectory while defining a different distribution through Eq. (1). Exactness at a particular τ is consequently not what the reported numbers rest on; what they require is that the reward *order* candidates as the posterior would, which is a weaker condition and the one measured in Section C.16. The conditions above matter for the positive-τ sampler, where the magnitudes of the weights, and not only their order, affect the outcome.

B.3 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*

$$\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t, \mathbf{y}) = \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t) + \left(\frac{1 - \alpha_t}{\sqrt{\alpha_t}}\right) \nabla_{\mathbf{x}_t} \log p_{Y|t}(\mathbf{y} \mid \mathbf{x}_t). \tag{10}$$

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

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

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. (10), we get

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

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

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

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*

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

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

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

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. The quantity

$$M_r(t) \triangleq \operatorname*{ess\,sup}_{p_{0|t,Y}(\cdot|\mathbf{x}_t, \mathbf{y})} r(X_0; \mathbf{s}) - \mathbb{E}[r(X_0; \mathbf{s}) \mid \mathbf{x}_t, \mathbf{y}] \in [0,1]$$

is the gap between the largest attainable reward and the mean reward under $p_{0|t,Y}(\cdot \mid \mathbf{x}_t, \mathbf{y})$.

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.

Remark 5. The three constants are nested, so the whole bound is controlled by $c_3(t) = \mathrm{Var}(X_0 \mid \mathbf{x}_t)$ alone. Since a variance is the smallest mean squared deviation from any constant and r is L_r-Lipschitz,

$$c_1(t) = \mathrm{Var}(r(X_0; \mathbf{s}) \mid \mathbf{x}_t, \mathbf{y}) \leq \mathbb{E}\left[\left(r(X_0; \mathbf{s}) - r(\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t, \mathbf{y}); \mathbf{s})\right)^2 \middle| \mathbf{x}_t, \mathbf{y}\right] \leq L_r^2\, c_2(t).$$

For the second link, write $\kappa_y \triangleq \sup_{\mathbf{x}_0} p_{Y|0}(\mathbf{y} \mid \mathbf{x}_0)/p_{Y|t}(\mathbf{y} \mid \mathbf{x}_t)$, which is the largest value of the density ratio f used in the proof above. Changing measure with f gives

$$c_2(t) \leq \mathbb{E}_{p_{0|t,Y}}[\|X_0 - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)\|_2^2] = \mathbb{E}_{p_{0|t}}[f(X_0)\|X_0 - \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)\|_2^2] \leq \kappa_y\, c_3(t).$$

Hence $c_1(t) \leq L_r^2 \kappa_y c_3(t)$ and $c_2(t) \leq \kappa_y c_3(t)$, and the bound of Proposition 3 becomes

$$|V_t^{\tau} - \hat{V}_t^{\tau}| \leq \frac{L_r}{\tau}\sqrt{c_3(t)}\left(\sqrt{\kappa_y} + \sqrt{c_4^{\eta}(t)}\right) + \min\left\{c_\tau L_r^2 \kappa_y\, c_3(t),\ \frac{M_r(t)}{\tau}\right\},$$

so at fixed τ the leading behaviour at small t is $O(\sqrt{c_3(t)}/\tau)$, the remaining term being $O(c_3(t))$. The reward is evaluated on $\hat{\mathbf{x}}_{0|t}$, so this is the regime the bound describes.

Remark 6. The quantity that governs the bound is moreover computable from the pre-trained denoiser, without drawing samples from $p_{0|t}$. The identity established in the proof above, $\left(\frac{1-\alpha_t}{\sqrt{\alpha_t}}\right)\nabla_{\mathbf{x}_t}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t) = \Sigma_{0|t}(\mathbf{x}_t)$, gives

$$c_3(t) = \operatorname{tr}\Sigma_{0|t}(\mathbf{x}_t) = \left(\frac{1-\alpha_t}{\sqrt{\alpha_t}}\right)\operatorname{tr}\big(\nabla_{\mathbf{x}_t}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)\big).$$

The trace on the right can be estimated with Hutchinson's identity [Hutchinson, 1989], $\operatorname{tr}(\mathbf{J}) = \mathbb{E}_{\mathbf{v}}[\mathbf{v}^T\mathbf{J}\mathbf{v}]$ for $\mathbf{v}$ with independent Rademacher entries, and each sample $\mathbf{J}\mathbf{v}$ is a single Jacobian-vector product through the denoiser network. A handful of such products therefore evaluates $c_3(t)$ along a trajectory, so the bound can be measured on a given run.

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

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

The same quantity admits a second bound. Bounding the integrand by its essential supremum gives $\log\mathbb{E}[\exp(r(X_0;\mathbf{s})/\tau)] \le \frac{1}{\tau}\operatorname{ess\,sup} r(X_0;\mathbf{s})$, and therefore

$$V_t^\tau(\mathbf{x}_t;\mathbf{s},\mathbf{y}) \le \frac{1}{\tau}\mathbb{E}[r(X_0;\mathbf{s})] + \frac{M_r(t)}{\tau}.$$

The statement retains the smaller of the two. With $c_\tau = e^{1/\tau} - 1 - 1/\tau$ and $M_r(t)/\tau \le 1/\tau$, the minimum is $O(1/\tau)$ as $\tau \to 0$, the same order in τ as the remaining terms. Then, we have

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

Now, let us simplify the first term,

$$\begin{aligned}
|\mathbb{E}[r(X_0;\mathbf{s})] - r(\tilde{\mathbf{x}}^\eta_{0|t,Y}(\mathbf{x}_t,\mathbf{y});\mathbf{s})| &\le \mathbb{E}[|r(X_0;\mathbf{s}) - r(\tilde{\mathbf{x}}^\eta_{0|t,Y}(\mathbf{x}_t,\mathbf{y});\mathbf{s})|] \\
&\le L_r\mathbb{E}[\|X_0 - \tilde{\mathbf{x}}^\eta_{0|t,Y}(\mathbf{x}_t,\mathbf{y})\|_2] \\
&\le L_r(\mathbb{E}[\|X_0 - \hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t,\mathbf{y})\|_2] + \|\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t,\mathbf{y}) - \tilde{\mathbf{x}}^\eta_{0|t,Y}(\mathbf{x}_t,\mathbf{y})\|_2).
\end{aligned}$$

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

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

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,

$$\begin{aligned}
\nabla_{\mathbf{x}_t}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t) &= \nabla_{\mathbf{x}_t}\int \mathbf{x}_0^T p_{0|t}(\mathbf{x}_0 \mid \mathbf{x}_t)\mathrm{d}\mathbf{x}_0 = \int \nabla_{\mathbf{x}_t} p_{0|t}(\mathbf{x}_0 \mid \mathbf{x}_t)\mathbf{x}_0^T\mathrm{d}\mathbf{x}_0 \\
&= \int \nabla_{\mathbf{x}_t}\log p_{0|t}(\mathbf{x}_0 \mid \mathbf{x}_t)\mathbf{x}_0^T p_{0|t}(\mathbf{x}_0 \mid \mathbf{x}_t)\mathrm{d}\mathbf{x}_0.
\end{aligned}$$

Now, we shall compute

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

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

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

which gives

$$\begin{aligned}\left(\frac{1-\alpha_t}{\sqrt{\alpha_t}}\right)\nabla_{\mathbf{x}_t}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t) &= \mathbb{E}_{X_0\sim p_{0|t}(\mathbf{x}_t)}[(X_0-\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))X_0^T] \\ &= \mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[X_0X_0^T]-\mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[X_0]\mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[X_0^T] \\ &= \mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[(X_0-\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))(X_0-\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))^T] \triangleq \Sigma_{0|t}(\mathbf{x}_t),\end{aligned}$$

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

Thus, we finally get,

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

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 $\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t,\mathbf{y}) = \mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[X_0f(X_0)]$. Now, we are ready to bound the final term as follows

$$\begin{aligned}&\|\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t,\mathbf{y})-\tilde{\mathbf{x}}^{\eta}_{0|t,Y}(\mathbf{x}_t,\mathbf{y})\|_2 \\ &\quad= \|\mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[X_0f(X_0)]-\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)-\eta\Sigma_{0|t}(\mathbf{x}_t)\mathbf{A}^T(\mathbf{y}-\mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))\|_2 \\ &\quad= \|\mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[X_0f(X_0)-\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)f(X_0)] \\ &\qquad\qquad -\eta\mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[(X_0-\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))(X_0-\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))^T]\mathbf{A}^T(\mathbf{y}-\mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))\|_2 \\ &\quad= \|\mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[(X_0-\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))(f(X_0)-\eta(X_0-\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))^T\mathbf{A}^T(\mathbf{y}-\mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))))]\|_2 \\ &\quad\le \mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[\|(X_0-\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))(f(X_0)-\eta(X_0-\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))^T\mathbf{A}^T(\mathbf{y}-\mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))))\|_2] \\ &\quad= \mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[\|X_0-\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)\|_2|f(X_0)-\eta(X_0-\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))^T\mathbf{A}^T(\mathbf{y}-\mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))|] \\ &\quad\le \sqrt{c_3(t)}\sqrt{c_4^{\eta}(t)},\end{aligned}$$

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

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

and at last, we have

$$\begin{aligned}c_4^{\eta}(t) &\triangleq \mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[(f(X_0)-\eta(X_0-\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))^T\mathbf{A}^T(\mathbf{y}-\mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)))^2] \\ &= \mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[f(X_0)^2]+\eta^2\mathbb{E}[((X_0-\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))^T\mathbf{A}^T(\mathbf{y}-\mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)))^2] \\ &\qquad -2\eta\mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[(X_0f(X_0)-f(X_0)\hat{\mathbf{x}}_{0|t}(t))^T]\mathbf{A}^T(\mathbf{y}-\mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)) \\ &= \mathbb{E}_{p_{0|t}(\mathbf{x}_t)}[f(X_0)^2]+\eta^2\|\mathbf{A}^T(\mathbf{y}-\mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t))\|^2_{\Sigma_{0|t}(\mathbf{x}_t)} \\ &\qquad -2\eta\langle\mathbf{A}(\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t,\mathbf{y})-\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)),\mathbf{y}-\mathbf{A}\hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)\rangle.\end{aligned}$$

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 $\mathbf{x}_t$.

B.3.1 Discussion on η

Proposition 3 states that the approximation error is small when t is small. In fact, under the (standard) assumptions, the error goes to 0 irrespective of the value of η. Thus, one can set $\eta = 0$ and still obtain consistent estimate of the conditional mean $\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t,\mathbf{y}) \approx \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)$ when t is small. Recalling that η stands in for $1/2\sigma_y^2$, the choice $\eta = 0$ is the $\sigma_y \to \infty$ limit, in which the measurement is treated as uninformative about $\mathbf{x}_0$ beyond what $\mathbf{x}_t$ already carries; the intuition below makes precise why this is harmless when t is small.

To get an intuitive justification from a simpler theoretical perspective, notice that the measurement $\mathbf{y} = \mathbf{A}\mathbf{x}_0 + \sigma_y\mathbf{z}$ is a noisy observation of $\mathbf{x}_0$ and $\mathbf{x}_t$ is a corruption of $\mathbf{x}_0$. When t is small, $\mathbf{x}_t$

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

Require: Side information $\mathbf{s}$, measurement $\mathbf{y}$, 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: **end for**
9: Select $\mathbf{x}_0^*$ from $\hat{\mathbf{x}}_{0|0}[1:N]$ (e.g., via reward maximization) and return $\mathbf{x}_0^*$

retains high-correlation with $\mathbf{x}_0$ and therefore contains more information about $\mathbf{x}_0$ than the noisy measurement $\mathbf{y}$. In this regime, additional conditioning by $\mathbf{y}$ contributes negligibly, and thus the approximation $\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t, \mathbf{y}) \approx \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t)$ holds. Formalizing the above notion, we notice that

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

and $\alpha_t \to 1$ as $t \to 0$, we have that

$$\hat{\mathbf{x}}_{0|t,Y}(\mathbf{x}_t, \mathbf{y}) \approx \hat{\mathbf{x}}_{0|t}(\mathbf{x}_t) + o_t(1).$$

Empirically, we compared $\eta = 1$ and $\eta = 0$ in inference-time search algorithm using DPS as the gradient in the second term is computed by DPS, and found that $\eta > 0$ does not provide additional performance gains. Since the other base samplers (MPGD, DAPS) do not provide the term involving gradient, we use $\eta = 0$ for these methods.

Finally, we remark that since the base sampler guides towards $\mathbf{y}$ explicitly from the likelihood term $\nabla_{\mathbf{x}_t} \log p_{t|Y}(\mathbf{x}_t)$, the generative process is gradually guided towards $\mathbf{y}$ and therefore, we eventually sample from $p_{0|Y,S}$ (as suggested by Proposition 1) and *not just* $p_{0|S}$.

B.4 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., 2025]. 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., 2025].

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].

C Additional Experiments

C.1 Perceptual Improvements Do Not Guarantee Better PSNR/LPIPS/SSIM

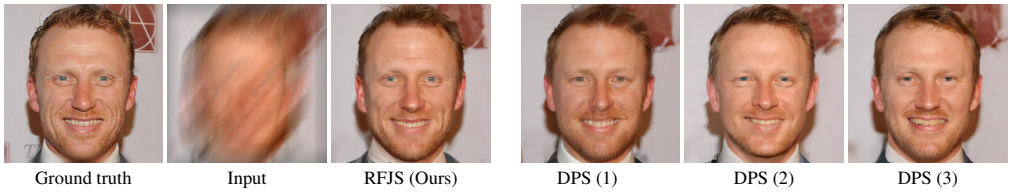


Figure 8: Qualitative illustration of the relevance of the FS metric and the superior performance of RFJS in identity preservation. RFJS reconstruction is clearly more faithful to the ground truth, yet PSNR, SSIM, and LPIPS values slightly favor the BlindDPS outputs.

Perceptual improvements do not necessarily lead to better values in classical metrics such as PSNR, LPIPS, and SSIM. Figure 8 provides a clear qualitative example: it compares three samples generated by BlindDPS with our RFJS reconstruction. Visually, our method preserves identity much better, and this is reflected in a lower FS. However, all three BlindDPS reconstructions achieve better PSNR, LPIPS, and SSIM than our result, showing that these traditional metrics can fail to capture semantic improvements.

We provide further evidence in Figure 9, where our reconstructions are perceptually closer to the ground truth than those of DPS, especially in challenging $32\times$ super-resolution settings. For these examples, Table 7 reports the standard metrics for our method and for reconstructions under DPS. Once again, the perceptual gains visible in the images are not fully reflected in PSNR, LPIPS, or SSIM. This is important because the main goal in inverse problems is to recover the underlying *semantics* of the images, while PSNR, LPIPS, and SSIM are only proxies for reconstruction quality. When a metric better aligns with perceptual quality, it should be preferred. In our case, for face reconstruction this is captured by the FS metric, and for ImageNet settings by CLIPScore. In both cases, our method yields significant improvements over the baselines in these perceptual rewards.

This discrepancy is also intuitive from how these metrics are defined. PSNR is a purely pixel-level measure: it rewards putting the "right" colors in the "right" locations, even if the high-level content is not faithfully preserved. LPIPS and SSIM operate on features extracted by neural networks or on low-level structural patterns, and are more sensitive to capturing the correct class and coarse structure than to fine-grained semantics. For example, when reconstructing a human face, these metrics are not strongly incentivized to verify the correct identity; they mainly check that the output still looks like a plausible face. This explains why they may remain in the same range even when perceptual quality clearly improves.

We emphasize that marginal differences in PSNR, SSIM, and LPIPS between our method and the baselines should not be interpreted as a weakness of our approach. As demonstrated above, these metrics are fundamentally misaligned with the goal of semantic reconstruction in ill-posed inverse problems, a conclusion supported by concurrent work [Lin et al., 2025, Su et al., 2025]. The relevant comparison is on task-specific perceptual metrics (FS and CLIPScore), where our method yields large and consistent improvements across all tasks and base samplers. We state the claim about the classical metrics in the form the evidence supports: search driven by a perceptual reward moves PSNR, SSIM and LPIPS within the same range, in either direction, rather than improving them systematically. The additional comparisons bear this out, and in both directions. Reward-guided beam search attains a higher PSNR than RFJS while trailing it on FS (Table 20), and ReF-LDM, a model trained for this task, is ahead on LPIPS and SSIM (Table 24). Section C.16 gives the particle-level reason: the reward orders candidates by identity and is close to uninformative about the distortion metrics, so there is no mechanism by which optimising it would move them in a fixed direction.

Other perceptual metrics. Table 6 extends Table 5 from three tasks to all twelve. DreamSim, the CLIP distance and the landmark error improve on every one of them, so the FS improvement is not an artefact of optimising an AdaFace score. They improve by less than FS does, which is what one

should expect when the reward is an identity model, and PSNR, LPIPS and SSIM improve on most tasks but not on all of them, as this section would predict.

Task	Algorithm	*General perceptual* DreamSim (↓)	CLIP (↓)	NME (↓)	*Task-specific and classical* FS (↓)	PSNR (↑)	LPIPS (↓)	SSIM (↑)
Default degradation								
Box Inpainting	DPS	0.078	0.088	0.0275	0.740	27.21	0.141	0.846
	RFJS (ours)	**0.054** (+30.8%)	**0.061** (+30.1%)	**0.0248** (+9.7%)	**0.298** (+59.8%)	**27.78**	**0.137**	**0.852**
Super Resolution	DPS	0.102	0.111	0.0235	1.054	24.47	0.232	0.684
	RFJS (ours)	**0.082** (+19.5%)	**0.090** (+19.0%)	**0.0232** (+1.2%)	**0.365** (+65.3%)	**24.57**	**0.227**	**0.686**
Gaussian Deblur	DPS	0.066	0.058	0.0222	0.808	25.55	0.202	0.703
	RFJS (ours)	**0.054** (+17.6%)	**0.047** (+18.6%)	**0.0212** (+4.7%)	**0.323** (+60.1%)	**25.61**	**0.198**	**0.706**
Motion Deblur	DPS	0.066	0.059	0.0223	0.819	25.85	0.196	0.726
	RFJS (ours)	**0.054** (+17.6%)	**0.044** (+24.8%)	**0.0214** (+4.0%)	**0.322** (+60.6%)	25.84	**0.192**	**0.728**
Non-linear Deblur	DPS	0.079	0.076	0.0236	1.007	23.49	0.234	0.659
	RFJS (ours)	**0.065** (+16.9%)	**0.058** (+24.4%)	**0.0236** (+0.2%)	**0.374** (+62.8%)	23.33	0.236	0.654
Blind Deblur	BlindDPS	0.069	0.061	0.0241	0.788	24.41	0.214	0.696
	RFJS (ours)	**0.058** (+15.5%)	**0.050** (+18.0%)	**0.0230** (+4.6%)	**0.339** (+57.0%)	**24.50**	**0.211**	**0.700**
Severe degradation (dps scale $= 0.8$)								
Box Inpainting 116^2	DPS	0.077	0.085	0.0292	0.697	23.40	0.143	0.835
	RFJS (ours)	**0.054** (+29.1%)	**0.056** (+34.1%)	**0.0267** (+8.8%)	**0.292** (+58.1%)	**24.08**	**0.139**	**0.841**
Box Inpainting 135^2	DPS	0.085	0.095	0.0307	0.815	21.54	0.164	0.800
	RFJS (ours)	**0.061** (+28.7%)	**0.066** (+30.5%)	**0.0281** (+8.2%)	**0.324** (+60.3%)	**21.72**	**0.159**	**0.805**
Super Resolution $\times 12$	DPS	0.137	0.151	0.0288	1.239	22.31	0.267	0.603
	RFJS (ours)	**0.101** (+26.3%)	**0.105** (+30.4%)	**0.0286** (+0.9%)	**0.443** (+64.2%)	**22.46**	**0.264**	**0.607**
Super Resolution $\times 16$	DPS	0.178	0.202	0.0331	1.310	20.98	0.296	0.563
	RFJS (ours)	**0.127** (+29.0%)	**0.136** (+32.7%)	**0.0318** (+4.1%)	**0.481** (+63.3%)	**21.04**	**0.293**	**0.567**
Super Resolution $\times 32$	DPS	0.291	0.330	0.0501	1.386	18.00	0.367	0.475
	RFJS (ours)	**0.210** (+27.8%)	**0.228** (+30.9%)	**0.0477** (+4.9%)	**0.558** (+59.7%)	**18.04**	**0.362**	**0.481**
Motion Deblur K=256	DPS	0.120	0.123	0.0292	1.212	22.18	0.263	0.604
	RFJS (ours)	**0.088** (+27.2%)	**0.085** (+30.4%)	**0.0271** (+7.0%)	**0.439** (+63.8%)	**22.20**	**0.261**	**0.607**

Table 6: Search with side information improves other perceptual metrics which are not directly derived from the reward modeling network, here over all twelve tasks: the full version of Table 5. RFJS attains the better value in every DreamSim, CLIP, NME and FS cell, and in most PSNR, LPIPS and SSIM cells. Percentages give the relative improvement of RFJS over DPS. The two degradation groups are not comparable across the group heading. Throughout, 64 images with $N = 8$ particles and a reference image as side information, with selection by the side-information reward alone; FAN detected a face in every ground truth and every reconstruction, so no sample is dropped.

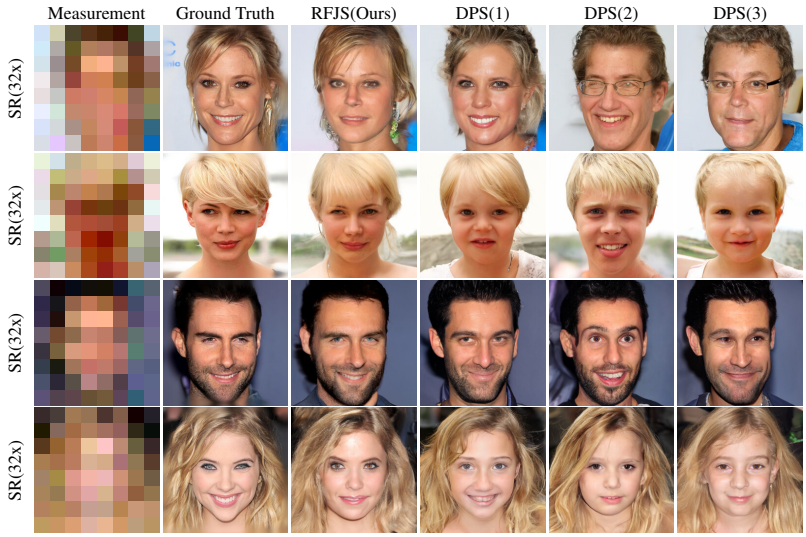


Figure 9: Our RFJS reconstructions preserve identity and perceptual details far better than DPS, as clearly visible from the images. This strong perceptual improvement is not reflected by classical metrics, which are reported in the table below.

Table 7: PSNR, LPIPS, and SSIM for the reconstructions shown in the figure above. Classical metrics fail to reflect the significant improvement.

Image ID	PSNR		LPIPS		SSIM	
	RFJS	DPS1	RFJS	DPS1	RFJS	DPS1
1	20.30	19.98	0.363	0.353	0.549	0.541
2	19.12	18.77	0.382	0.413	0.567	0.552
3	20.95	20.98	0.254	0.273	0.609	0.596
4	17.85	18.54	0.411	0.406	0.368	0.389
Avg	**19.56**	**19.57**	**0.353**	**0.361**	**0.523**	**0.520**

C.2 DPS

We also present additional qualitative examples for our DPS experiments in Figures 10 and 11, corresponding to settings that use face and text as side information, respectively.

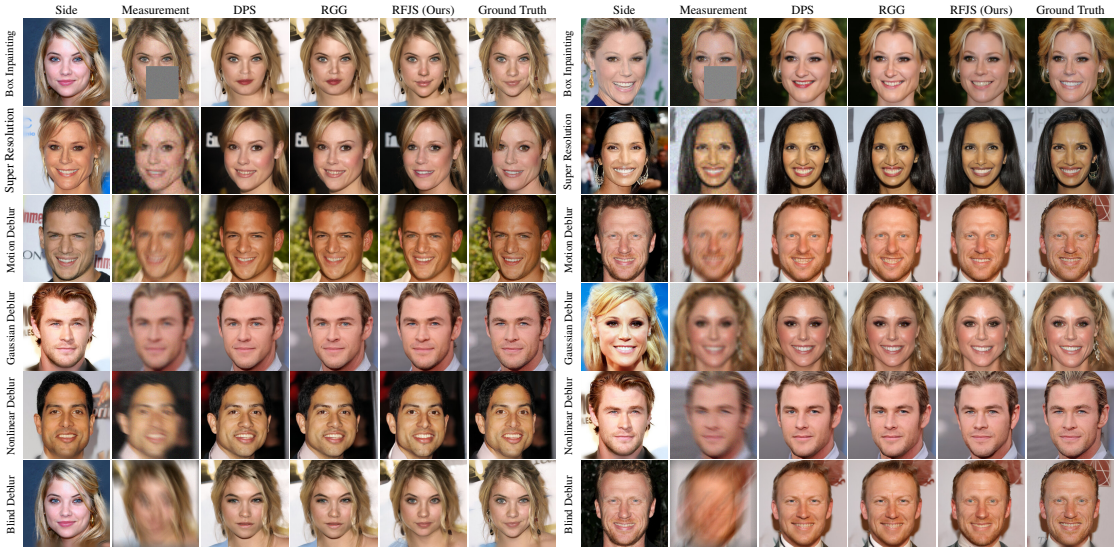


Figure 10: Additional samples using DPS as the base sampler.

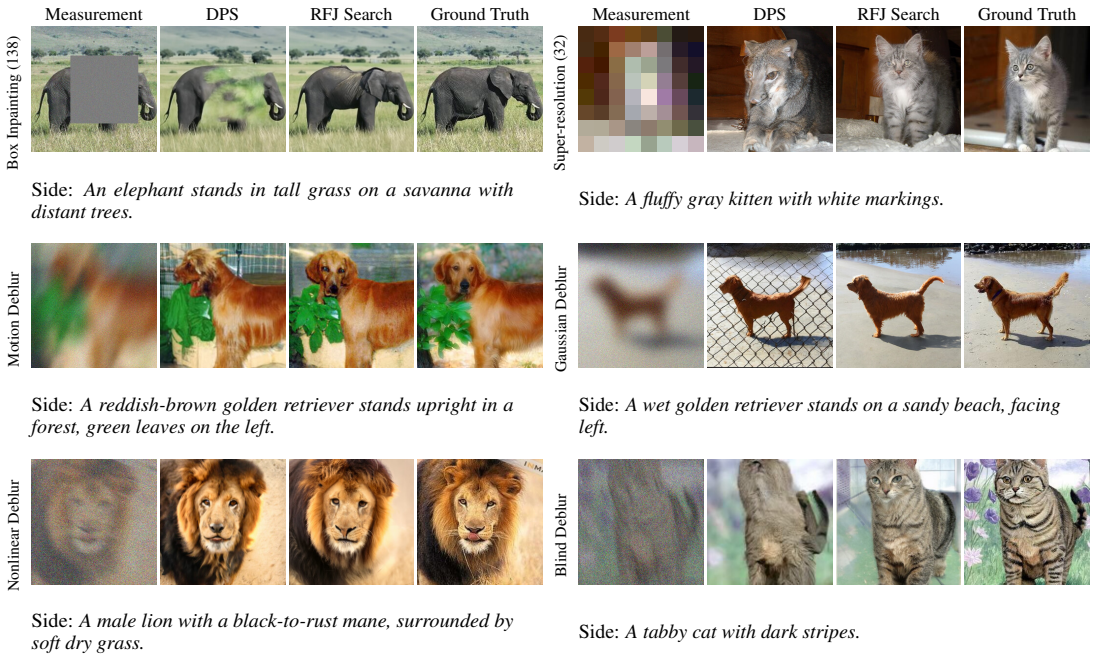


Figure 11: Qualitative comparison on ImageNet with textual side information. For highly degraded measurements, DPS and BlindDPS often produce artifacts, whereas our method reduces these by better aligning reconstructions with the description.

C.3 Expanded Text Evaluation on 75 Images

The textual side-information results of Table 3 are computed on the 25 ImageNet images described in Section 5. To check that they do not depend on that particular sample, we repeat the whole evaluation on an expanded set of 75 images that contains the original ones as a subset, leaving every other setting unchanged ($N = 4$ particles, resampling base $B = 100$; see Section D.2). We also add DreamSim [Fu et al., 2023], a perceptual distance fit to human similarity judgements, as a fifth metric; like PSNR, SSIM and LPIPS it is computed against the ground truth and takes no part in selection, which uses the side-information reward alone. Table 8 reports the result.

The conclusions of the main text carry over to the larger set. RFJS and GS lead the base solver on the metrics that track agreement with the side information or with human perceptual judgement – CS, LPIPS and DreamSim – on all six tasks, with DreamSim improving by between 12% and 31% and CS by between 2% and 9%. PSNR and SSIM stay in the same range as the baseline, better on some tasks and level on the rest, and worse on none. This is the pattern Section C.1 describes: search concentrates mass on reconstructions that agree with $\mathbf{s}$, which a pixel-wise average does not reward.

Against Best-of-N, which draws the same N samples and keeps the one the reward prefers but never resamples during the trajectory, RFJS is ahead on CS and DreamSim on most of the tasks, so the gain comes from resampling along the trajectory and not from the final reward-based pick alone. Gaussian deblur is the exception: there Best-of-N attains the better PSNR and SSIM while the two are level on CS.

RFJS and GS are close to each other throughout this set. We therefore read the expanded text evaluation as evidence for reward-guided search as such, rather than for the recursive fork–join schedule specifically over a greedy one. The two separate more clearly when the reward is sharper: with face identity as side information, RFJS attains a lower face-similarity distance than GS on every task in Table 1.

	Box Inpainting					Super Resolution (×32)				
Algorithm	CS (↑)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	DS (↓)	CS (↑)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	DS (↓)
RFJS (ours)	**0.892**	**18.80**	**0.636**	**0.301**	**0.177**	**0.805**	15.39	**0.281**	**0.520**	**0.320**
GS (ours)	0.890	18.59	0.635	0.305	0.182	0.798	**15.45**	0.278	0.523	0.336
BoN (w/ side)	0.883	18.70	0.634	0.308	0.191	0.771	15.35	0.271	0.526	0.367
DPS	0.861	18.23	0.629	0.317	0.233	0.739	15.38	0.271	0.535	0.419
	Gaussian Deblur					Motion Deblur				
Algorithm	CS (↑)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	DS (↓)	CS (↑)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	DS (↓)
RFJS (ours)	**0.843**	16.17	0.294	0.468	0.249	**0.865**	17.13	0.344	**0.441**	**0.202**
GS (ours)	0.834	16.34	0.297	0.468	0.250	0.860	16.73	0.327	0.453	0.212
BoN (w/ side)	0.841	**16.82**	**0.319**	**0.454**	**0.243**	0.848	**17.29**	**0.346**	0.446	0.219
DPS	0.788	15.66	0.285	0.493	0.337	0.810	16.59	0.322	0.472	0.293
	Non-linear Deblur					Blind Deblur				
Algorithm	CS (↑)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	DS (↓)	CS (↑)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	DS (↓)
RFJS (ours)	0.874	18.52	0.410	0.410	0.195	**0.894**	19.94	0.494	**0.354**	**0.138**
GS (ours)	0.871	**18.64**	0.409	**0.408**	0.191	0.890	20.01	0.497	**0.354**	0.140
BoN (w/ side)	**0.876**	18.58	**0.413**	0.414	**0.190**	0.884	**20.11**	0.498	0.355	0.148
DPS / BlindDPS	0.854	18.55	**0.413**	0.420	0.222	0.873	20.07	**0.502**	0.367	0.169

Table 8: **Text as side information,** 75 **images.** The evaluation of Table 3 repeated on a three-times-larger set, with DreamSim (DS) added. Bold is best and underline second best within each task; values that tie at the reported precision share the same emphasis. The baseline row is DPS for the first five tasks and BlindDPS for blind deblur. RFJS and GS lead on CS and DreamSim on every task, while PSNR and SSIM are led by Best-of-N on several of them.

C.4 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., 2025]. 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.

Results: The qualitative results are given in Figure 12, while the quantitative metrics are given in 9. Observe that for the task of inpainting, our search algorithm shows a significant improvement over DAPS, particularly in the FS metric.

Table 9: 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 underlined. We observe that our RFJ Search-based algorithm has the best or the second-best performance in all the tasks.

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

C.5 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

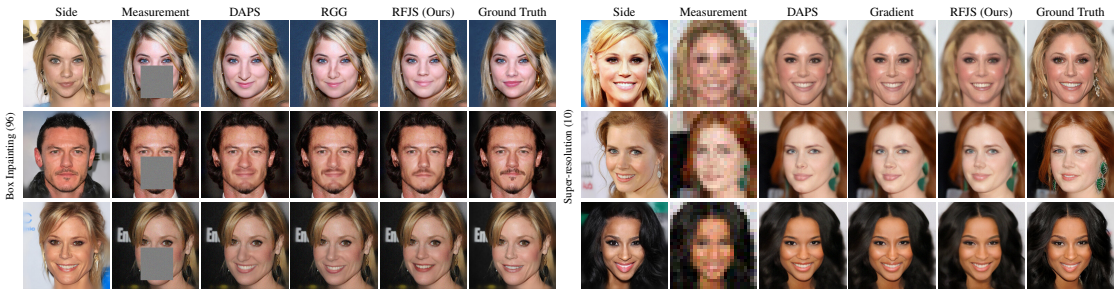


Figure 12: Qualitative comparison of algorithms using DAPS as the base sampler. Our method offers better reconstructions aligned with the identity.

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 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 C.8.

Results: Experimental results with MPGD [He et al., 2024] as baseline algorithm are given in Table 10. We observe that using RFJ Search significantly enhances the FS metric, while improving the other metrics. The reconstructions that utilize the side information exhibit strong identity match with the ground truth, which is reflected in the FS metric.

Table 10: 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 RFJ search 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.

Sampler	Task	Method	FS (↓)	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

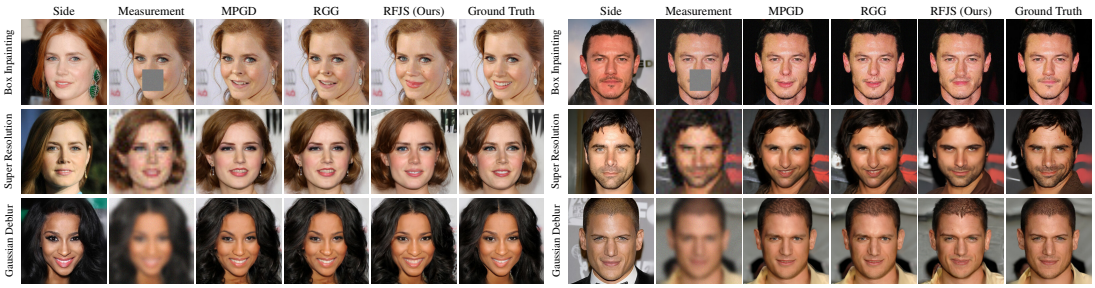


Figure 13: Qualitative comparison of algorithms using MPGD as the base sampler. Our method offers better reconstructions aligned with the identity. Notice that while the identity is preserved, the exact reconstruction might not be possible as witnessed in, for example, Box inpainting. The ground truth 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 FS improvements are significant.

C.6 Stable Diffusion

To further demonstrate the generality of our framework, we evaluate it on Stable Diffusion, a large-scale latent diffusion model that natively conditions on text. A natural baseline in this setting is to simply pass the textual side information as the input prompt to Stable Diffusion and run the standard inverse problem solver (DAPS), which directly leverages the model's native text-conditioning mechanism without any modification. We compare this baseline against our inference-time search applied on top of the same text-conditioned solver, where particles are selected using a reward that scores consistency between reconstructions and the textual side information.

Even though Stable Diffusion already incorporates the text prompt through its native conditioning, we find that performing inference-time search over the resulting trajectories yields further improvements. This indicates that text-conditioning alone does not fully exploit the information contained in the side description: search allows the solver to explore multiple plausible reconstructions consistent with the prompt and select those that best align with the side information through the reward.

We additionally compare against a Best-of-N variant that uses the same number of particles but selects the final reconstruction based on the measurement residual $\|\mathbf{y} - \mathbf{A}\mathbf{x}_0^i\|$ rather than the side-information reward. This measurement-based selection strategy fails to improve, and in noisy or severely ill-posed regimes can actively degrade, the reconstruction quality, since it favors particles that explain the noise rather than the underlying signal. In contrast, our reward-based selection using side information consistently improves all metrics across box inpainting, super-resolution, motion deblur, and Gaussian deblur tasks.

Qualitative reconstructions are shown in Figure 14, where our method recovers perceptually faithful content that aligns with the textual description, while the DAPS baseline often produces artifacts or content inconsistent with the prompt. Quantitative results are reported in Table 11, which confirms that selection by our reward $r(\mathbf{x}_0; \mathbf{s})$ yields consistent gains over both the base sampler and Best-of-N without side information. Together, these results show that our framework is complementary to native conditioning mechanisms and provides additional gains even when the base model is already trained to consume the same modality of side information.

	Box Inpainting				Gaussian Deblur			
Algorithm	PSNR (↑)	LPIPS (↓)	SSIM (↑)	CS (↑)	PSNR (↑)	LPIPS (↓)	SSIM (↑)	CS (↑)
BON (w/ SI, ours)	**19.51**	**0.262**	**0.750**	**0.893**	**20.30**	**0.527**	**0.494**	**0.789**
BON (w/o SI)	18.02	0.274	0.738	0.846	<u>20.05</u>	<u>0.540</u>	<u>0.487</u>	<u>0.748</u>
DAPS	<u>18.21</u>	<u>0.274</u>	<u>0.742</u>	<u>0.855</u>	19.86	0.540	0.481	0.735
	Super Resolution				Motion Deblur			
Algorithm	PSNR (↑)	LPIPS (↓)	SSIM (↑)	CS (↑)	PSNR (↑)	LPIPS (↓)	SSIM (↑)	CS (↑)
BON (w/ SI, ours)	**18.65**	**0.579**	**0.398**	**0.767**	**19.54**	**0.551**	**0.474**	**0.770**
BON (w/o SI)	18.13	0.590	0.368	<u>0.729</u>	18.18	0.562	0.462	0.718
DAPS	<u>18.44</u>	<u>0.583</u>	<u>0.391</u>	0.727	<u>19.49</u>	<u>0.556</u>	<u>0.466</u>	<u>0.722</u>

Table 11: Without side information, Best-of-N selection by $\|\mathbf{y} - \mathbf{A}\mathbf{x}_0^i\|$ fails to improve over the baseline and can degrade it. With side information, selection by our reward $r(\mathbf{x}_0; \mathbf{s})$ consistently improves all metrics (Box Inpainting and Super Resolution are reproduced from the table provided in the section Role of side information in the main paper).

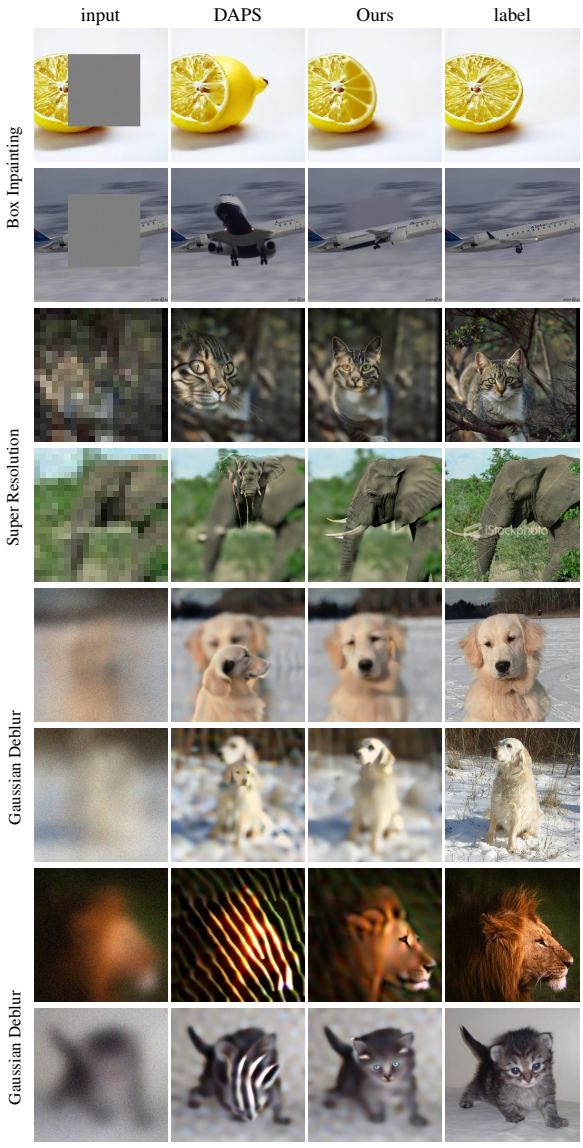


Figure 14: Stable Diffusion

C.7 Ablation Studies

In this section, we conduct ablation studies to study the performance and robustness of our algorithm by degrading: (i) Measurement quality and (ii) Side information quality.

C.7.1 Measurement Quality: Hard Inverse Problems

In this section, we demonstrate the effectiveness of our method on hard inverse problems with severely degraded measurements. For DPS, we evaluate inpainting with a large mask that covers almost the entire face, super-resolution with aggressive downsampling ratios of 12× and 32×, and motion deblurring using a large 256-pixel kernel. For MPGD, we similarly consider challenging variants of inpainting, super-resolution, and Gaussian deblurring, with quantitative results reported in Tables 12 and 13. Across all settings, our method consistently improves the target reward metric, which translates directly into stronger qualitative performance, and we also observe improvements in classical metrics on average. The qualitative results are shown in Figures 15, 16, 9. For textual side information, we already include challenging experimental settings in the main paper; see Figure 5 and Table 3.

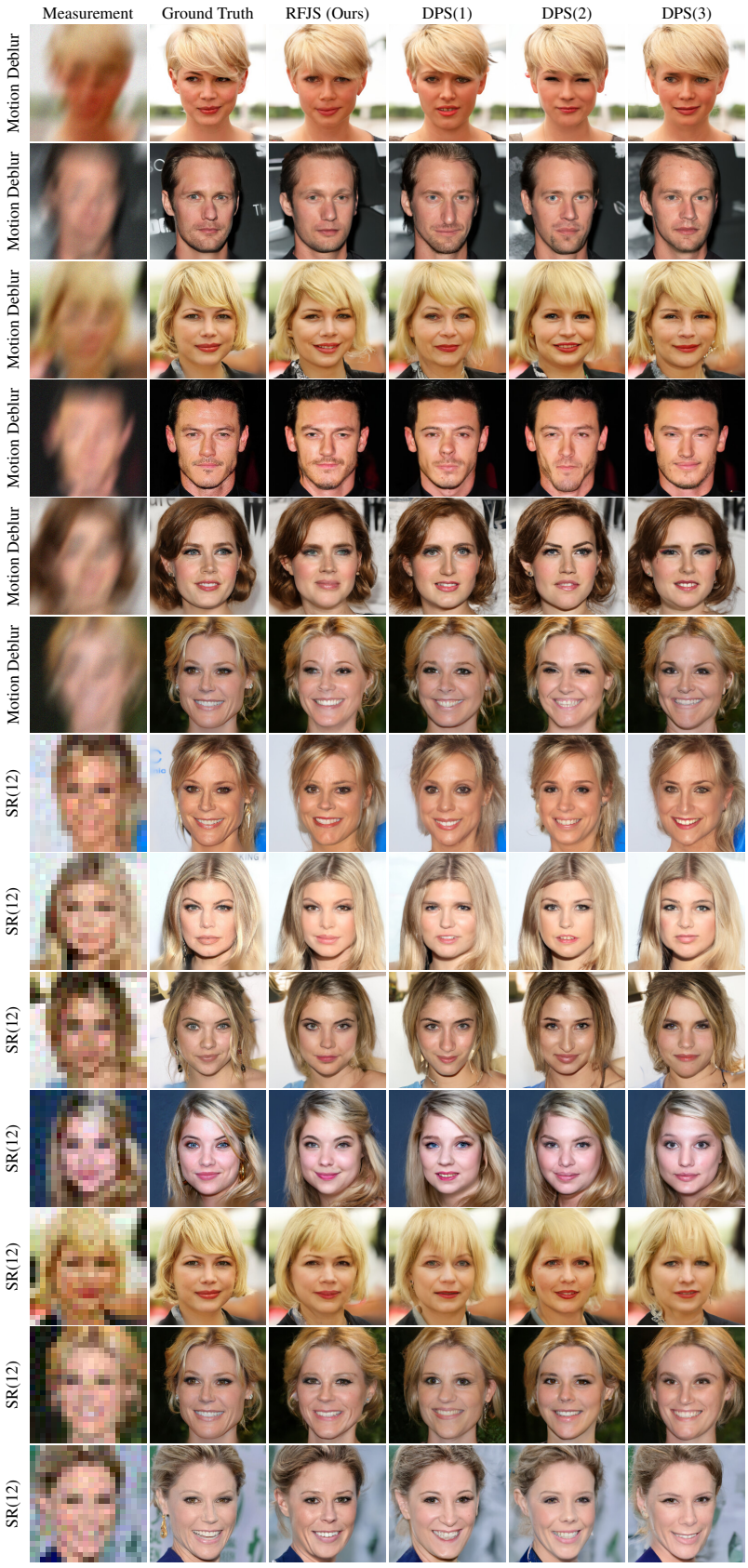


Figure 15: Qualitative results on hard tasks using DPS. Our algorithms reconstructs faces with perceptual quality.

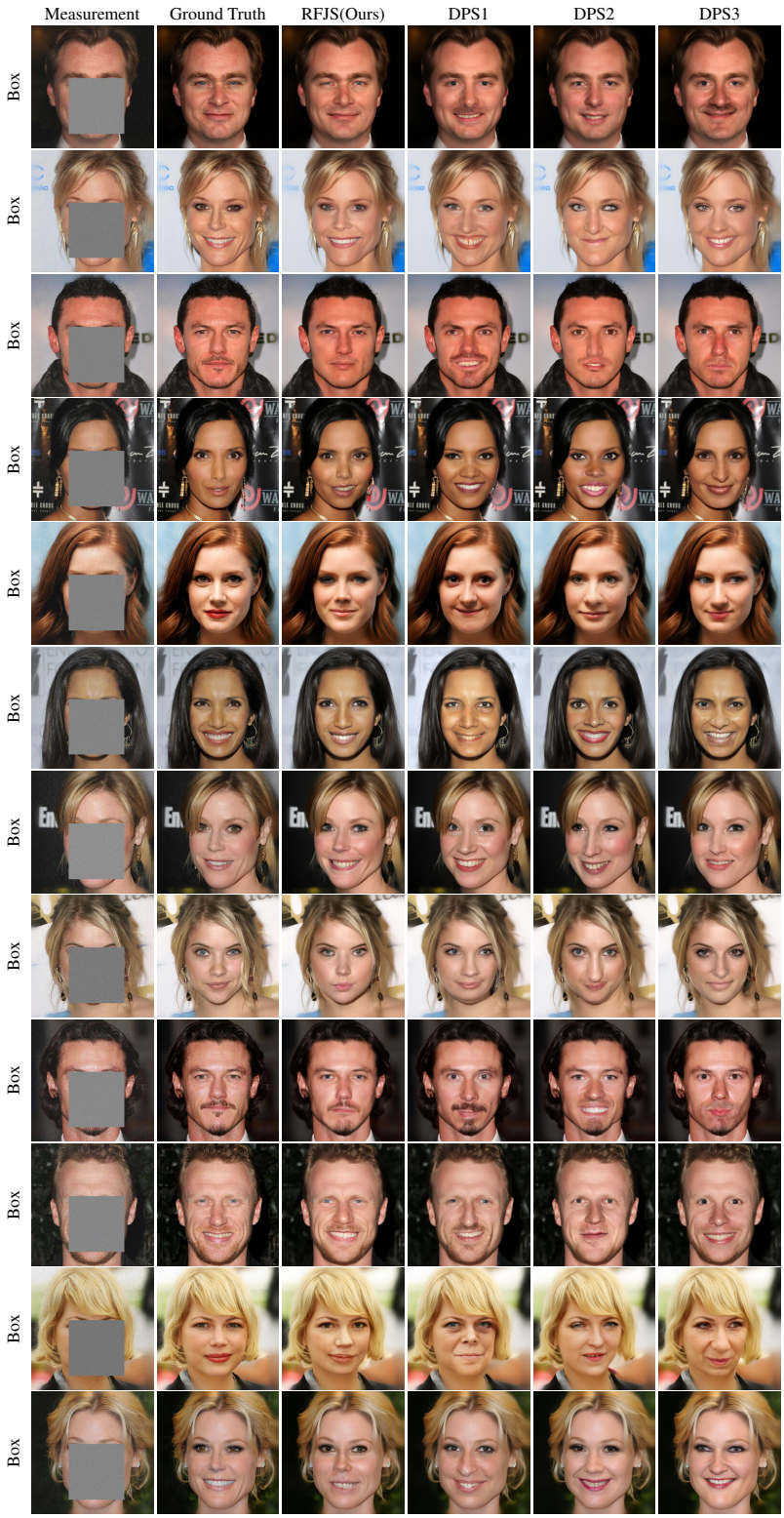


Figure 16: Qualitative results on box inpainting using DPS sampler. Since the face is completely masked in the measurement, side information provides useful clues so that the reconstruction preserves identity.

Table 12: Quantitative results using DPS for hard tasks with $\text{dps scale} = 0.8$ (guidance strength from the measurement y)

Task Name	Task Parameters	Algo	FS (↓)	PSNR (↑)	LPIPS (↓)	SSIM (↑)
Box Inpainting	$M = 116$	RFJS	**0.47**	**26.62**	**0.140**	**0.839**
		DPS	1.07	26.21	0.151	0.826
Super Resolution	$S = 12$	RFJS	**0.603**	**22.92**	**0.262**	**0.619**
		DPS	1.23	22.78	0.265	0.613
	$S = 32$	RFJS	**0.748**	**18.66**	**0.354**	**0.496**
		DPS	1.38	18.43	0.363	0.489
Motion Deblur	$K = 256$	RFJS	**0.545**	22.67	**0.257**	0.614
		DPS	1.21	**22.69**	0.260	**0.615**

Table 13: Quantitative results using MPGD for hard tasks with $\text{scale} = 6.0$ (guidance strength from the measurement y)

Task Name	Task Parameters	Algo	FS (↓)	PSNR (↑)	LPIPS (↓)	SSIM (↑)
Inpainting	$M = 96$	RFJS	**0.594**	**24.83**	**0.171**	**0.749**
		MPGD	0.831	24.79	0.172	0.747
Super Resolution	$S = 12$	RFJS	**1.032**	**21.40**	**0.312**	**0.555**
		MPGD	1.282	21.32	0.318	0.553
	$S = 16$	RFJS	**1.142**	**19.03**	0.383	**0.486**
		MPGD	1.356	19.01	**0.381**	0.486
Gaussian Deblur	$K = 81, I = 5.0$	RFJS	**0.820**	**24.49**	**0.238**	**0.645**
		MPGD	0.993	24.39	0.241	0.641

C.7.2 Side Information Quality

In this subsection, we evaluate how the quality of side information affects the performance of our method. For the face experiments, we study a $4\times$ super-resolution task using DPS as the base sampler. We degrade the side information by applying a Gaussian blur with a 31×31 kernel and varying blur intensities. Figure 17 reports the resulting FS scores. As expected, performance consistently improves as the side information becomes more reliable. The case with blur intensity 0.0 corresponds to the unaltered side information and matches the results reported in the main paper. Representative qualitative examples are provided in Figure 18. Notably, even with heavily degraded side information (intensity 5.0), our method (FS: 0.53) outperforms the baseline (FS: 1.042), and the performance steadily improves as the fidelity of the side information increases.

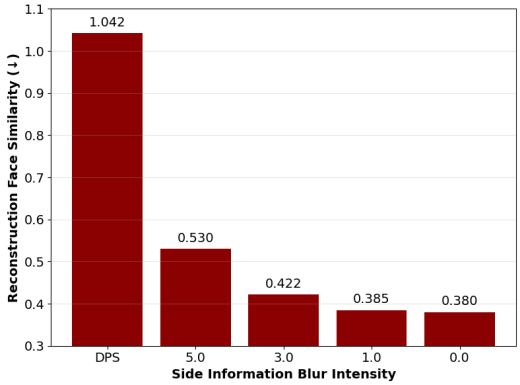


Figure 17: Performance of RFJS algorithm with side information blurred with Gaussian kernel in the face data experiments

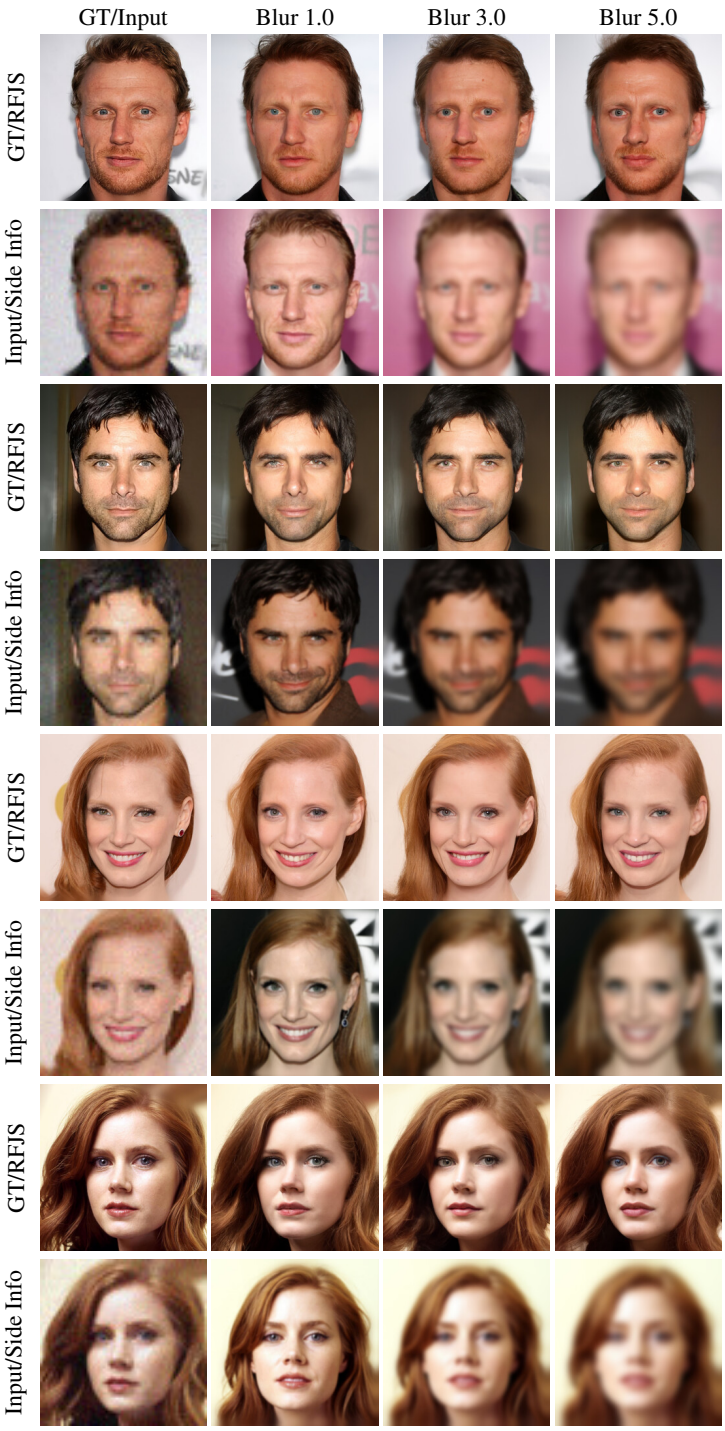


Figure 18: Qualitative comparison of generations with side information blurred at different levels. Some sharp features that preserve identity are lost when the side information is severely blurred. In the last three columns, first rows indicate reconstruction and second rows indicate the blurred side information that is used to guide the generation process.

For the case of textual side information, Figure 19 shows that our method remains consistent under small variations in the prompt. In this example, the key missing information in the measurement is the type of animal or object present in the scene. Once the side information specifies that there is a golden retriever, the reconstruction improves significantly. Further modifying the prompt does not

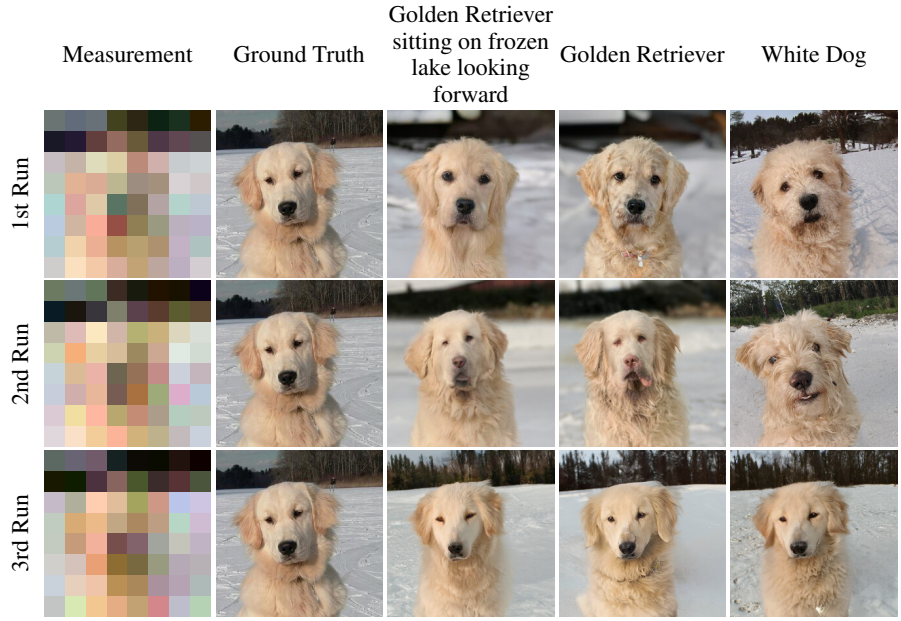


Figure 19: Qualitative comparison of generations with different textual descriptions as side information. Due to randomness and robustness of the algorithm due to relative ordering, the quality assessment is difficult.

substantially affect the output. If the side information is "white dog", then our algorithm reconstructs a dog which is not necessarily a golden retriever (Figure 19, last column).

Semantically incorrect side information. The degradation study above keeps the side information correct and reduces its fidelity. A different and more adversarial question is what happens when the side information is simply wrong, that is, when it carries confident information about the wrong target. We test this on $4\times$ super-resolution by replacing the reference image of each target with an image of a *different* identity, leaving everything else unchanged. Table 14 reports the result. Identity fidelity degrades sharply, which is the expected and correct behaviour, but the distortion metrics move very little. The reason is structural: the search selects among candidates produced by the measurement-conditioned transition kernel of the base solver, and never overrides $\mathbf{y}$. An unreliable reward can therefore cause the algorithm to select the wrong identity among reconstructions that all remain plausible under the measurement, but it does not let the reconstruction drift away from the measurement altogether. In a severely ill-posed problem, where several semantically distinct reconstructions explain the same $\mathbf{y}$, this is a genuine failure mode, and we discuss it further in Section E.

Side information	FS↓	PSNR↑	LPIPS↓	SSIM↑
Correct reference	**0.53**	**27.5**	**0.175**	**0.773**
Reference of a different identity	0.91	27.1	0.183	0.759

Table 14: Effect of semantically incorrect side information on $4\times$ super-resolution with DPS as the base solver. Replacing the reference with an image of a different identity costs most of the identity gain, while the distortion metrics are almost unchanged, because the search only chooses among measurement-consistent candidates. The run uses $N = 8$ particles, $B = 16$ and a DPS step size of 0.8, and is evaluated on 8 images; the absolute values are therefore not comparable with the $4\times$ super-resolution row of Table 1, and only the comparison between the two rows here is meaningful.

Multiple reference images. Table 15 compares the two aggregations of Section 4.1 for $K = 2$, on box inpainting with a 116×116 mask, DPS as the base solver, $N = 8$ particles and $B = 16$. The gain over a single reference is consistent but modest: the mean improves three of the four metrics and the maximum improves all four. That ordering is what one expects when the two references differ in

how closely they match the pose and lighting of the target, since rewarding alignment to the closest reference then ignores the less informative one.

Reference set	PSNR↑	LPIPS↓	SSIM↑	FS↓
$K = 1$	26.11	0.1389	0.8451	0.2750
$K = 2$, mean reward	26.42	0.1404	0.8491	0.2696
$K = 2$, max reward (nearest reference)	**26.58**	**0.1337**	**0.8498**	**0.2613**

Table 15: Aggregating the rewards of several reference images, for box inpainting with a 116×116 mask, DPS as the base solver, $N = 8$ particles and $B = 16$, evaluated on 8 images.

C.8 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 20, 21, 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 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 FS metric while leaving the other evaluation metrics in the same range and avoiding artifacts (see Figure 21). Moreover, the sensitivity to gradient scale increases when the number of reverse diffusion steps is small. For instance, in DAPS (Figure 20) 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 22, and 23.

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 with fewer steps (e.g., 100 steps in MPGD) exhibit increased sensitivity to the gradient scale, as illustrated in Figures 22 and 23.

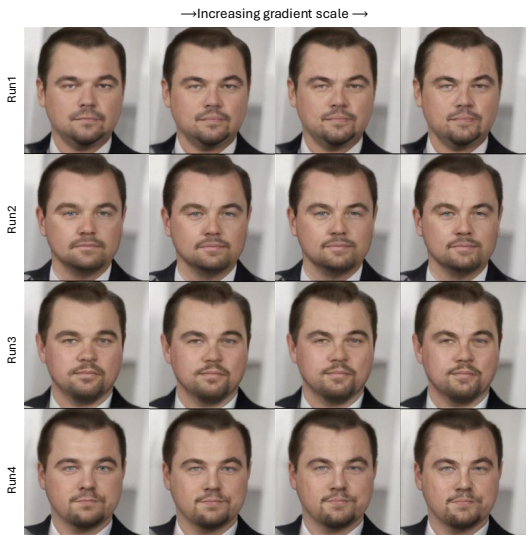


Figure 20: **Effect of reward-gradient guidance in diffusion-based inverse problems.** We show 4 runs with different random seeds (rows), and for each seed we vary the gradient scale across 4 settings (columns). Within each row, the noise realization is identical and only the gradient scale changes; 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.

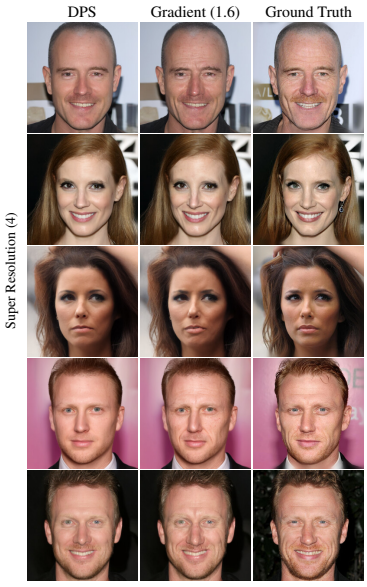


Figure 21: Qualitative comparison of the effect of gradient scale on reconstruction paths.

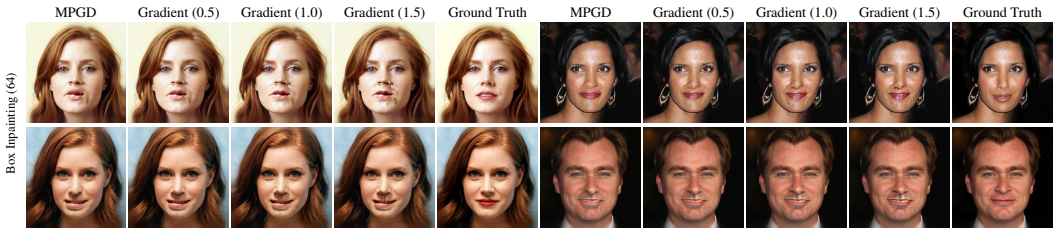


Figure 22: Qualitative comparison of the effect of gradient scale on reconstruction paths. Notice that while the base reconstruction is reasonable, adding the gradient can degrade it if the scale is very large. The final scale used in the experiments is 0.5.

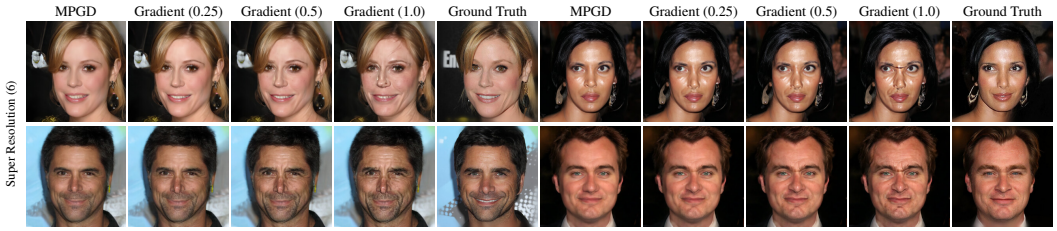


Figure 23: Qualitative comparison of the effect of gradient scale on reconstruction paths for super resolution task. Notice that while the base reconstruction is reasonable, adding the gradient can degrade it if the scale is very large. The final scale used in the experiments is 0.25.

C.9 Effect of Number of Particles

In this section, we study how performance scales with the number of particles. Figure 24 presents the results for the box inpainting task using RFJS with DPS as the baseline sampler, while Figure 25 reports the corresponding results for the remaining tasks. The evaluations using MPGD as the baseline are shown in Figure 26. Across all tasks and for both baseline samplers, we observe a consistent improvement in performance as the number of particles N increases, aligning with the expected behavior of particle-based search methods. Furthermore, our results indicate that RFJS scales more efficiently with increasing N than both Greedy Search and Best-of-N.

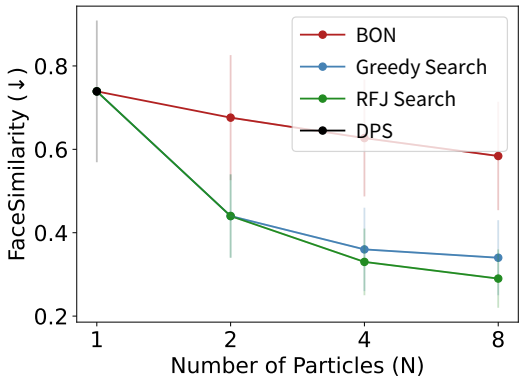
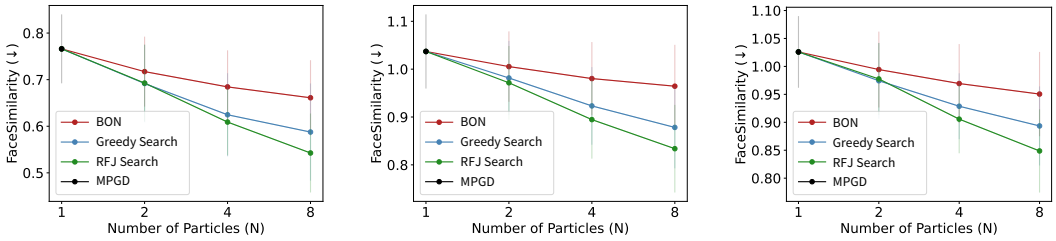
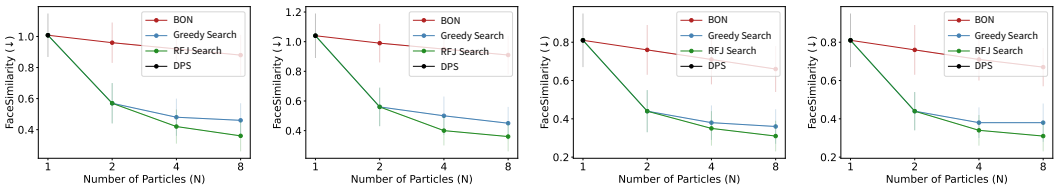


Figure 24: Scaling of search algorithms with respect to the number of particles.



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

Figure 26: Effect of number of particles N on the FS 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.



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

Figure 25: Effect of the number of particles N on the FS metric in DPS. As N increases, the performance improves.

C.10 Runtimes

Table 16 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.

Finally, resampling of the particles requires computing the rewards, which involves call to the reward evaluation model. Therefore, decreasing the base B (more frequent resampling) increases the wall-clock time. Denoting c_d, c_r the computation cost per function call to diffusion model and reward model respectively, a rough estimate of the time taken to run the algorithm is given by:

$$N(O(Tc_d) + O((T/B)c_r)) \leq N(O(Tc_d) + O(Tc_r)),$$

where T denotes the number of diffusion steps and N the number of particles. However, it is important to note that the wall-clock time is dominated by the call to the diffusion model, which is a much larger network than the reward model, i.e, $c_d \gg c_r$ in practice, and so the reward evaluation does not add a significant overhead.

The compute–quality tradeoff is a feature, not a limitation. Our framework provides a controllable knob: increasing N yields better reconstructions, while $N = 1$ recovers the base sampler exactly at no additional cost. This is precisely the inference-time scaling behavior that has driven recent progress in LLMs [Snell et al., 2025, Setlur et al., 2025], and is a desirable property rather than a drawback. We also note that any additional runtime cost is *inherent to inference-time search as a paradigm*, it is not specific to our method. BON, Greedy Search, and any other particle-based approach share the same linear scaling with N. What distinguishes our framework is that the added compute is spent meaningfully: as shown in Fig. 1 (right), scaling N without side information yields

negligible gains, whereas our reward-guided search produces consistent and significant improvements with every additional particle. **The runtime cost is therefore justified by gains that no amount of unguided compute scaling can achieve.**

Table 16: 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 $N = 1$.

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

C.11 Ablation on Reward Function

Choice of reward function. A natural question is how to choose the reward function $r(x_0; s)$ and how sensitive the framework is to this choice. As discussed in Section 4.1, the reward function need only satisfy a simple and intuitive condition: it should assign high values to pairs (x_0, s) that are consistent, and low values to inconsistent pairs. Any pretrained model that captures this notion of pairwise consistency is a valid candidate. In practice, such models are widely available for common side-information modalities, face recognition networks for image side information, and text-image alignment models for textual side information, and require no additional training.

To identify the best candidate for each modality, we conducted a brief literature review of pretrained models known to capture the relevant notion of consistency in prior recognition or generative modeling settings. For face side information, we evaluated AdaFace Kim et al. [2022] and FaceNet, both trained to assign higher similarity to image pairs from the same identity. For textual side information, we evaluated ImageReward Xu et al. [2023] and CLIP score Radford et al. [2021], where the former is trained to reflect human preference and text-image alignment, and the latter uses a contrastive objective for matched image-text pairs.

We note that during diffusion sampling, the reward is evaluated on intermediate denoised estimates $\hat{x}_{0|t}(x_t)$ (the Tweedie estimate) rather than directly on the noisy state x_t, since $\hat{x}_{0|t}$ lies closer to the image manifold and therefore falls more reliably within the distribution on which these networks were trained. Robustness to such intermediate inputs was an additional practical consideration in our selection.

Crucially, *all reward candidates we tested improved over the baseline*, confirming that the framework is not sensitive to the specific choice of reward, as long as it satisfies the consistency condition above. We ultimately selected AdaFace and ImageReward as they yielded the strongest empirical performance. The full comparison is reported in Tables 17 and 18.

Clarification on optimization target vs. evaluation metric. We wish to explicitly address a potential source of confusion. Our method optimizes the reward $r(\hat{x}_0; s)$, which measures consistency between the *reconstructed image* $\hat{x}_0$ and the *side information* s. However, the FS metric reported in our tables measures the distance between the *reconstructed image* $\hat{x}_0$ and the *ground truth* x_0^*.

These are **not the same quantity**. The ground truth x_0^* is never observed at inference time and plays no role in the reward computation or the search procedure. The side information s is a *different* image of the same identity, taken under different conditions (pose, lighting, etc.), and is not equal to x_0^*. So the improvements we get are not a consequence of optimizing the same quantity that is being measured, and it is not reward hacking.

Below, we provide an ablation on choice of reward functions for face as side information and text as side information.

Task	Param	Algo	FS↓	PSNR↑	LPIPS↓	SSIM↑
Box Inpainting	M=96	DPS	0.739	27.93	0.139	0.852
		FaceNet	0.534	26.55	0.144	0.852
		AdaFace	**0.308**	**28.29**	**0.136**	**0.855**
Super Resolution	S=12	DPS	1.23	22.78	0.265	0.613
		FaceNet	0.962	22.69	0.265	0.608
		AdaFace	**0.603**	**22.92**	**0.262**	**0.619**
Motion Deblur	K=256	DPS	1.21	**22.69**	0.260	**0.615**
		FaceNet	0.945	21.88	0.277	0.587
		AdaFace	**0.545**	22.67	**0.257**	0.614

Table 17: Comparison of RFJS with different reward functions for image side information. FS is evaluated against the *ground truth* (not the side information used during inference). Both FaceNet and AdaFace improve over the DPS baseline, confirming that any reward satisfying the consistency condition is beneficial; AdaFace achieves the best overall performance and is used in the main paper.

Task	Param	Algo	CS↑	PSNR↑	LPIPS↓	SSIM↑
Box Inpainting	M=138	DPS	0.866	19.87	0.314	0.667
		CLIP	0.875	19.93	0.305	0.670
		ImageReward	**0.895**	**20.75**	**0.290**	**0.681**
Super Resolution	S=32	DPS	0.737	16.83	0.521	0.326
		CLIP	0.784	16.37	0.511	0.311
		ImageReward	**0.814**	**17.10**	**0.493**	**0.343**
Motion Deblur	K=256	DPS	0.783	17.70	0.468	0.364
		CLIP	0.836	17.60	0.465	0.353
		ImageReward	**0.866**	**18.27**	**0.439**	**0.367**

Table 18: Comparison of RFJS with different reward functions for textual side information. CLIP-Score is evaluated against the *ground truth* image (not the text prompt used during inference). Both CLIP-based and ImageReward-based rewards improve over the DPS baseline; ImageReward achieves the best performance and is used in the main paper.

C.12 Hyperparameter Sensitivity

Our framework introduces very few hyperparameters. The temperature τ controls the softness of particle resampling, but as discussed in Section 4.2, RFJS naturally balances exploration and exploitation through its hierarchical grouping structure, because of this, in practice, we set τ to a sufficiently small value so that resampling becomes deterministic (always retaining the highest-reward particle within each group), eliminating τ as a tuning parameter entirely.

The primary remaining hyperparameter is the resampling base B. We conduct an ablation study over $B \in \{4, 8, 16, 32\}$ on the box inpainting task using DPS as the base sampler, with $N = 8$ particles. The results are reported in Table 19. Across all values of B, RFJS substantially improves the perceptual FS metric over the DPS baseline (from 0.739 to a range of 0.296–0.434), while keeping classical metrics (PSNR, LPIPS, SSIM) in a consistently better range. Crucially, the variation across choices of B is small, confirming that the method is not sensitive to this hyperparameter.

In practice, we select B once per base sampler, reflecting properties of the sampler such as its number of diffusion steps and degree of inherent exploration, and apply the same value uniformly across all tasks using that sampler. No per-task tuning of B is performed.

Method	PSNR↑	LPIPS↓	SSIM↑	FS↓
DPS (baseline)	27.929	0.1391	0.8519	0.739
RFJS (B=4)	28.360	0.1365	**0.8562**	**0.296**
RFJS (B=8)	28.186	**0.1354**	0.8551	0.330
RFJS (B=16)	**28.385**	0.1363	**0.8562**	0.372
RFJS (B=32)	28.264	0.1383	0.8553	0.434

Table 19: Ablation over resampling base B for RFJS with DPS on box inpainting (N=8, M=96). All values of B substantially improve the perceptual FS metric over the DPS baseline, and classical metrics remain in a consistently better range across all choices, demonstrating low sensitivity to B.

C.13 Evaluating Search Strategies in a 2D Setting

To illustrate the effect of side information in inference-time search, we consider a simple 2D setup. The prior on x_0 is a mixture of Gaussians, shown in the leftmost image of Figure 28. A ground-truth sample x_0 is drawn from this prior, and a random forward operator A is generated with fixed norm and random orientation. The measurement is then $y = Ax_0 + n$. The corresponding posterior $p(x_0 \mid y)$, approximated by DPS, is also shown in Figure 28. Because this is an ill-posed inverse problem, the posterior is multimodal and DPS fails to reconstruct the correct solution.

We then add side information of the form $s = Dx_0 + n_s$, where D is chosen so that Dx_0 is orthogonal to Ax_0. The reward function is defined as $r(x_t, s) = -\|s - D\hat{x}_0(x_t)\|_2^2$, where $\hat{x}_0(x_t)$ is the Tweedie estimate of the clean signal from the noisy state x_t. Using this reward, RFJ Search produces posterior samples $p(x_0 \mid y, s)$ as shown in the three rightmost images of Figure 28. This is the Gaussian case of Section B.2: because $\mathbf{s}$ is generated as a noisy linear measurement of x_0 and the reward is the negative squared residual, the reward-tilted target of Eq. (1) coincides with $p(x_0 \mid y, s)$ at $\tau = 2\sigma_s^2$, so this setting is one of the few in which the particles can be compared against the distribution they are meant to represent. As the resampling base B decreases, the particles concentrate more tightly around the ground truth, highlighting how side information guides inference.

To compare RFJ and Greedy Search, we next consider a more realistic scenario where side information is generated through a neural network reward model. For example, in face reconstruction tasks, side information may come from the embedding of an additional image, while for text-conditioned tasks, it may come from a text encoder. To mimic this, in each trial we sample a ground truth x_0, a forward operator A, and a reward network r_θ with randomly initialized weights. The side information is defined as $s = r_\theta(x_0) + n_s$, and the reward is the cosine similarity between $r_\theta(\hat{x}_0(x_t))$ and s. We repeat this experiment 16,000 times, and for each trial run both RFJ and Greedy Search 128 times with $N = 8$ particles, varying the resampling base B over powers of two from 1 to 256. The average PSNR values are reported in Figure 27. Across all settings, RFJ Search consistently outperforms Greedy Search, with the best performance observed at $B = 4$.

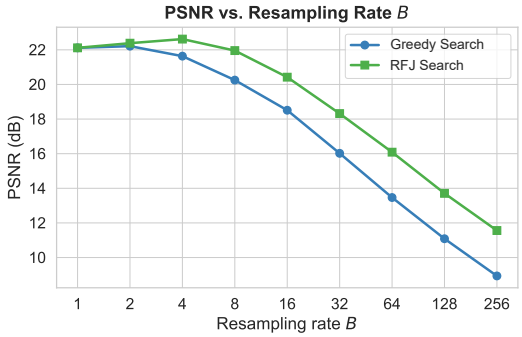


Figure 27: Comparison of performance of RFJ Search (RFJS) and Greedy Search (GS) as a function of B for a randomly generated reward network r_θ. RFJS outperforms GS across all values of B.

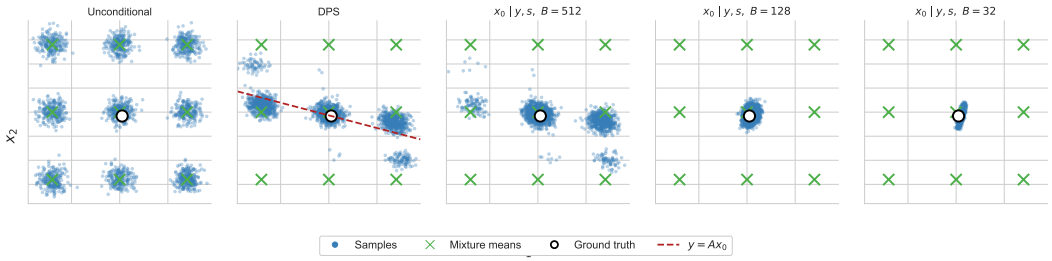


Figure 28: Illustration of the effect of B in utilization of side information for the reconstruction in a linear inverse problem with a mixture of Gaussian prior.

C.14 Alternative Resampling and SMC Strategies

Section A places our algorithms on the two axes that specify an SMC method, weighting and resampling. The strategies compared here each vary one of them. *Adaptive SMC* replaces the fixed resampling clock with a data-driven trigger, resampling whenever the effective sample size $1/\sum_i w_i^2$ falls below a threshold. *Annealed SMC* tempers the weights over time through $\exp(\lambda_t r/\tau)$ with λ_t increasing from 0 to 1, so that selection pressure is weak early in the trajectory and sharpens as the estimate of $\mathbf{x}_0$ becomes reliable. *Stratified resampling* reduces the variance of the survivor draw by partitioning $[0, 1]$ into equal bins and drawing one index per bin; *multinomial resampling* is the textbook alternative, included as its reference point. *Adaptive group sizes* retains the structure of RFJS but sets the group size from a run-time statistic rather than a fixed schedule. *Reward-guided beam search* maintains M beams, expands each into K children and deterministically keeps the top M by reward.

Setting. All strategies solve the same box-inpainting task with a 96×96 mask and DPS as the base solver, use $N = 8$ particles and the same AdaFace reward, and are evaluated on the same 16 images; they differ only in the selection rule. The resampling cadence is $B = 16$ for RFJS, adaptive group sizes and beam search, and $B = 4$, which resamples the smallest group at every reverse step, for the SMC strategies, which is their textbook setting. Adaptive SMC uses systematic resampling whenever the effective sample size falls below $N/2$, with a temperature given by a robust (median-absolute-deviation) estimate of the spread of the reward increments; we swept this temperature over 1, 2, 4 and $16\times$ and report $2\times$. Annealed SMC uses the same weights with a linear ramp $\lambda_t = t/T$ and resamples at every step. The stratified and multinomial rows both resample globally from weights proportional to $\exp(r/\tau)$ with the same robust τ, and differ only in how the survivors are drawn. Adaptive group sizes uses the $B = 16$ and the group range 2–8 of RFJS, but sets g_t to the nearest power of two to the run-time effective sample size. Beam search keeps $M = 2$ beams of width $K = 4$ and prunes deterministically at the same cadence.

Search algorithm	PSNR↑	LPIPS↓	SSIM↑	FS↓
RFJS (ours)	<u>28.23</u>	**0.128**	**0.877**	**0.196**
Adaptive SMC	28.01	0.142	0.857	0.254
Annealed SMC	27.54	0.143	0.853	0.218
Stratified resampling	27.83	0.143	0.850	0.209
Multinomial resampling	27.62	0.156	0.842	0.213
Adaptive group sizes	28.02	<u>0.139</u>	<u>0.864</u>	0.211
Reward-guided beam search	**28.36**	<u>0.139</u>	0.863	<u>0.206</u>

Table 20: Alternative weighting and resampling rules on box inpainting with a 96×96 mask, DPS as the base solver and $N = 8$ particles, evaluated on 16 images. Every strategy uses the side information through the same AdaFace reward and differs only in the selection rule (**best**, <u>second best</u>). RFJS attains the best LPIPS, SSIM and FS, and the second-best PSNR.

Method	PSNR↑	LPIPS↓	SSIM↑	FS↓
DPS (no resampling)	25.72	0.1403	0.8437	1.2250
Stochastic resampling, $\tau = 1/3$	25.97	**0.1368**	0.8456	0.5142
Stochastic resampling, $\tau = 1/10$	**26.67**	0.1372	**0.8467**	**0.2996**

Table 21: Stochastic resampling at a positive temperature, for box inpainting with a 116×116 mask, DPS as the base solver, $N = 8$ particles and $B = 16$, evaluated on 8 images. Positive-temperature resampling is a direct implementation of the reward-tilted formulation of Proposition 1. Both temperatures improve FS substantially over the base sampler while also improving PSNR, SSIM and LPIPS, and identity preservation improves as τ decreases; the deterministic selection used throughout our experiments is the endpoint of that trend.

C.15 Additional Baselines

An SMC base solver: MCGdiff. Because our framework acts on top of a base sampler, it can also be applied to a sampler that already maintains particles. We augment MCGdiff [Cardoso et al., 2024] by replacing its resampling rule every fourth step with resampling based on our side-information

reward, and evaluate on $32\times$ super-resolution. Table 22 reports the result. Adding RFJS improves MCGdiff on all four metrics. The gain in FS is smaller than the corresponding gain for DPS, which we attribute to the fact that the particles of MCGdiff do not evolve independently between side-information resampling steps, so there is less trajectory diversity for the search to exploit. The table also contains the direct comparison between MCGdiff and DPS with RFJS.

Method	PSNR↑	LPIPS↓	SSIM↑	FS↓
MCGdiff ($N = 64$)	17.980	0.3825	0.4868	1.3970
MCGdiff + RFJS ($N = 64$)	18.207	0.3709	<u>0.4938</u>	<u>0.9821</u>
DPS ($N = 8$)	18.025	<u>0.3667</u>	0.4831	1.3997
DPS + RFJS ($N = 8$)	**18.426**	**0.3574**	**0.4853**	**0.3931**

Table 22: RFJS applied on top of MCGdiff, an SMC solver, for $32\times$ super-resolution with face side information. The side-information reward replaces the resampling rule of MCGdiff every fourth step. Each method is run at its own particle budget, $N = 64$ for MCGdiff and $N = 8$ for DPS, and all four rows are evaluated on the same 16 images.

A text-conditioned base solver: TReg. Section C.6 already evaluates our framework with Stable Diffusion as the base model. Here we instead use TReg [Kim et al., 2025a] as the base solver and apply our search on top of it, for the box-inpainting task with textual side information. Selection is Best-of-N over $N = 8$ particles, that is, the $B \geq T$ endpoint of Greedy Search, in which the particles evolve independently and the reward is applied once at the end of the trajectory. Table 23 shows that the search improves every reported metric relative to TReg alone.

Method	PSNR↑	LPIPS↓	SSIM↑	CS↑
TReg	15.25	0.393	0.542	0.814
TReg + ours	**15.81**	**0.381**	**0.550**	**0.838**

Table 23: Our search applied on top of TReg, a text-conditioned solver, for box inpainting with a 138×138 mask and textual side information. Selection is Best-of-N over $N = 8$ particles, evaluated on 24 ImageNet images.

A trained reference-conditioned model: ReF-LDM. ReF-LDM [Hsiao et al., 2024] is a latent diffusion model trained for reference-based face restoration, and therefore provides a point of comparison in which the side information is used by a trained conditional model rather than at inference. We compare it against DPS with RFJS and $N = 8$ particles on super-resolution at a mild and a severe degradation level, and on motion deblurring. Both methods receive the same degraded measurement and the same single reference image. Table 24 reports the results. At the degradations for which ReF-LDM was trained it is better on LPIPS and SSIM, while our training-free search is better on PSNR and on FS. On motion deblurring, a degradation for which ReF-LDM was not trained, RFJS is better on every metric and by a large margin.

These numbers should not be read as isolating the mechanism by which side information is used, because the two approaches differ in several components at once. ReF-LDM uses a stronger latent diffusion model with a learned VQ decoder, whereas our method uses an off-the-shelf pixel-space FFHQ prior, and the sharper outputs encouraged by the learned decoder may account for part of its advantage on LPIPS and SSIM. Our method incorporates the measurement through an inverse-problem solver at inference time, whereas ReF-LDM is trained on particular degradation processes, which is what the motion-deblurring row reflects. Finally, ReF-LDM learns to incorporate the reference during training, whereas our framework introduces it only at inference through a reward.

Task	Method	PSNR↑	LPIPS↓	SSIM↑	FS↓
Super Resolution, S=4	ReF-LDM (trained)	24.89	**0.183**	**0.755**	0.253
	RFJS (ours, training-free)	**25.04**	0.218	0.702	**0.241**
Super Resolution, S=12	ReF-LDM (trained)	19.94	**0.267**	**0.592**	0.445
	RFJS (ours, training-free)	**20.74**	0.294	0.566	**0.327**
Motion Deblur	ReF-LDM (trained)	22.25	0.204	0.688	0.292
	RFJS (ours, training-free)	**26.48**	**0.188**	**0.752**	**0.208**

Table 24: Comparison against ReF-LDM, a model trained for reference-based face restoration. Both methods receive the same measurement and the same single reference image. Motion deblurring is a degradation outside the training distribution of ReF-LDM. The RFJS rows use $N = 8$ particles and $B = 16$, and both methods are evaluated on the same 8 images.

C.16 Reward Calibration and Ranking Fidelity

Section B.2 records the condition under which the reward-tilted target of Eq. (1) coincides with the posterior: the reward must be an affine function of the log-likelihood, $r(\mathbf{x}_0; \mathbf{s})/\tau = \log p_{S|0}(\mathbf{s} \mid \mathbf{x}_0) + c(\mathbf{s})$ with c free of $\mathbf{x}_0$. This subsection examines how closely the reward we actually use satisfies that condition, and, since the main experiments operate in the $\tau \to 0$ limit in which only the ordering of the reward matters, how reliably that ordering agrees with an oracle that has access to the ground truth.

Calibration. Of the three rewards we use, AdaFace is the one whose training objective leaves magnitudes free (Section B.2), so it is the one worth measuring. Equation (8) is what makes the measurement possible. It writes the log-likelihood ratio that Eq. (2) is a statement about, $\log\big(p_{S|0}(\mathbf{s} \mid \mathbf{x}_0)/p_S(\mathbf{s})\big)$, in closed form as $\log M + \log\langle \pi(\mathbf{x}_0), \pi(\mathbf{s})\rangle$, so the quantity the reward is meant to be affine in can be evaluated for any pair from the two embeddings alone, with no density to estimate. We evaluate it at $\kappa = 8$ on the pairs the search actually forms, which are pairs of a reconstruction with a photograph of the same person: $\mathbf{x}_0$ ranges over the 512 reconstructions of the $\times 16$ super-resolution run of Table 1 and each is paired with both photographs of its own identity, giving 1,024 pairs. On the same pairs we evaluate the reward we use, $r(\mathbf{x}_0; \mathbf{s}) = -\|\Phi(\mathbf{x}_0) - \Phi(\mathbf{s})\|_2$, and fit a straight line of the first quantity on the second by least squares. Equation (2) is the statement that this line fits, with slope $1/\tau$, and we measure how well it does by R^2: one minus the residual sum of squares divided by the total sum of squares of the log-likelihood ratio about its mean, over the same 1,024 pairs. Two features of that figure fix how it should be read. It is a fit to the individual pairs, with nothing averaged beforehand; grouping the pairs into bins by r and fitting the bin means instead discards the scatter within each bin and reports a markedly higher value on the same data. And its denominator is the spread of the log-likelihood ratio over these pairs alone, so it describes how well the line predicts a single pair drawn from the regime the search operates in, rather than how well the reward separates photographs of different people, which lie further apart and are correspondingly easier to tell apart. The line accounts for 80% of that variation, a correlation of 0.90, at a slope corresponding to $\tau = 0.46$. The embedding distance is therefore an approximately affine image of the exact quantity Eq. (8) supplies, informative about it rather than equivalent to it, which is what the $\tau \to 0$ limit asks of it: the reward enters only through the ordering it induces on the particles of one image.

Ranking fidelity. In the $\tau \to 0$ limit the algorithm reads nothing from the reward but the ordering it induces: at a resampling step it holds $N = 8$ candidate reconstructions of the same image and keeps whichever the reward scores highest, so the only thing that matters is whether the reward orders those eight as an observer with access to the ground truth would. We measure that agreement directly. For each image we rank the eight particles by the reward $r(\mathbf{x}_0; \mathbf{s})$, which sees only the side information, and rank them again by an oracle metric computed against the ground truth. Kendall's τ is the fraction of the $\binom{8}{2} = 28$ particle pairs that the two rankings place in the same relative order, minus the fraction they place in opposite orders: $\tau = 1$ means the two rankings are identical, $\tau = 0$ that they agree no more often than chance, and $\tau = -1$ that one is the reverse of the other. Table 25 reports this averaged over the 64 evaluation images, for four choices of oracle, with DPS as the base solver.

The FS column is a control rather than a result: that oracle is the same embedding distance as the reward, measured against the ground truth instead of against $\mathbf{s}$, and since the two images are photographs of one person the rankings ought to agree. They do, at 0.76 to 0.88, decreasing as the inverse problem becomes more ill-posed and more reconstructions become consistent with the measurement. The other three columns carry the content. Against the distortion oracles the agreement is near zero, so at the moment of selection the reward is almost uninformative about which particle will score better on PSNR, LPIPS or SSIM. This is the particle-level counterpart of the effect documented in Section C.1: a search driven by a perceptual reward moves the perceptual metric decisively and leaves the distortion metrics in roughly the same range, because at the point where it chooses it cannot tell them apart.

	Kendall's τ against oracle			
Task	**FS**	**LPIPS**	**PSNR**	**SSIM**
Box Inpainting, M=96	0.876	0.163	0.171	0.177
Box Inpainting, M=116	0.874	0.076	−0.011	0.122
Box Inpainting, M=135	0.874	0.137	0.073	0.118
Super Resolution, S=4	0.840	−0.022	0.051	0.037
Super Resolution, S=12	0.801	0.099	0.047	0.054
Super Resolution, S=16	0.767	0.050	0.073	0.084
Super Resolution, S=32	0.762	0.037	0.002	0.000
Gaussian Deblur	0.857	0.041	−0.011	0.022
Motion Deblur, K=61	0.853	0.067	0.023	0.081
Motion Deblur, K=256	0.830	0.040	0.102	0.086
Nonlinear Deblur	0.848	0.098	0.048	0.084
Blind Deblur	0.798	0.182	0.176	0.198

Table 25: Agreement between the ranking of the $N = 8$ particles induced by the AdaFace reward and the ranking induced by an oracle with access to the ground truth, averaged over the 64 evaluation images, with DPS as the base solver. Kendall's τ is 1 for identical rankings and 0 for chance agreement. FS is a control, being the reward's own distance measured against the ground truth; the remaining columns show that the reward is nearly uninformative about the distortion metrics.

D 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.

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

D.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.

D.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, measurement noise $\sigma = 0.05$ (the default).
- **Super-resolution:** Downsampling rate $32\times$, measurement noise $\sigma = 0.05$ (the default).
- **Motion/Gaussian deblur:** Kernel size 256, intensity 12.0 for the Gaussian kernel and 0.5 for the motion kernel, measurement noise $\sigma = 0.1$.
- **Nonlinear deblur:** Kernel size unchanged, measurement noise $\sigma = 0.5$.
- **Blind deblur:** Kernel size 64 with intensity 0.5, measurement noise $\sigma = 0.02$. This task is run with BlindDPS, which estimates the kernel jointly with the image, and its absolute scores are therefore not comparable with the five DPS tasks above.

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

Three further settings are shared across these tasks. Resampling is enabled only after the first 400 of the 1000 diffusion steps, so that the particles diverge before any trajectory is duplicated. The reward-gradient scale is zero throughout, so the text results isolate the contribution of search and contain no reward gradient guidance. The DPS measurement-gradient step size is 0.5 for box inpainting, 0.9 for super-resolution and 0.3 for each of the four deblurring tasks.

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.

D.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.

The closest method to ours in this setting is TGVN [Atalık et al., 2025], which constrains the ambiguous subspaces of the solution using a second contrast through a learned unrolled network, and from which we take the contrast pairings above. We report no quantitative comparison against it: the released repository provides training code and data splits but no trained weights, so reproducing it means training from scratch, and the authors' configuration for a single experiment requests four A100 GPUs for up to fourteen days on fastMRI data that requires credentialed access.

E 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.

Table 26: Quantitative MRI reconstruction results (fastMRI knee, AF=16, ACS=2%).

Algorithm	PSNR (↑)	SSIM (↑)	LPIPS (↓)	NMI (↑)
	PDFS with PD			
RFJS	**25.85**	**0.801**	**0.375**	**0.457**
GS	25.33	<u>0.797</u>	<u>0.375</u>	<u>0.455</u>
BON	<u>25.47</u>	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	<u>27.80</u>	<u>0.920</u>	<u>0.360</u>	<u>0.579</u>
BON	27.80	0.918	0.366	0.570
ContextMRI	27.46	0.915	0.375	0.563

Misleading side information. Our formulation assumes that $\mathbf{s}$ comes from an external source that carries genuine information about $\mathbf{x}_0$, such as a second photograph of the same person, a complementary MRI contrast, or a description written by someone who saw the scene. When that assumption fails, the method inherits the failure. The mechanism is a wrong reward ranking: in a severely ill-posed problem several semantically distinct reconstructions explain the same measurement, and a reward computed against misleading side information will rank the wrong one highest. We make no claim to eliminate hallucination. What the structure of the algorithm does bound is the extent of the failure, since the search only selects among candidates produced by the measurement-conditioned kernel of the base solver and never overrides $\mathbf{y}$; Table 14 measures this, with identity fidelity degrading sharply while the distortion metrics stay close to their correct-reference values. The stakes are higher in safety-critical settings. In medical imaging in particular, a confidently wrong auxiliary contrast would produce a reconstruction that is consistent with the acquired data and still clinically misleading, which is the reason for treating the output as decision support rather than as a measurement.

Compute for quality. Maintaining N particles costs compute, and the quality we report is bought with it. The analytical cost is given in Section C.10 and grows linearly in N, but the wall-clock cost grows considerably more slowly because the particles are processed in parallel: on box inpainting with DPS, one particle takes 55 seconds per image and $N = 2$, 4 and 8 take 75, 131 and 241 seconds, so eight particles cost about $4.4\times$ the baseline rather than $8\times$. The trade-off is a knob rather than a fixed price. Setting $N = 1$ recovers the base solver exactly, and the gains do not require the largest budget: $N = 2$ already removes roughly 40% of the FS error on box inpainting, from 0.74 to 0.44 (Section C.9). Still, for applications where inference time is tightly constrained, the compute overhead is a genuine limitation of inference-time search as a family of methods.

Broader impacts. Our method can improve reconstruction quality in scientific imaging applications such as MRI, with positive impact on diagnostic imaging. Potential negative impacts include increased fidelity of identity-conditioned reconstructions, which raises privacy and consent considerations. In safety-critical domains such as medical imaging, our method should be treated as decision support only, with domain-expert review.

F 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.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state our three main contributions (modeling side information via reward-tilted posteriors, two inference-time search algorithms, and extensive experiments across multiple solvers and modalities), and these are supported by the theoretical results in Section 4 and the experiments in Section 5 and Appendix C.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include a dedicated Limitations section (Appendix E) discussing the absence of formal optimality guarantees for our exploration-exploitation strategies and the reliance on principled heuristics and tuned schedules.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Proposition 1 and Proposition 3 are stated with all assumptions in the main paper, and complete proofs are provided in Appendix B.1 and Appendix B.3 respectively.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 5 describes the experimental setup, baselines, and evaluation metrics. Appendix D provides detailed hyperparameter settings (number of particles, resampling base, gradient scales, noise levels, kernel sizes) for each task, sampler, and side-information modality. We also provide our code via the link in the abstract.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide a public code repository (linked in the abstract) with instructions to reproduce the main experiments. All datasets used (CelebA-HQ, FFHQ, ImageNet, fastMRI) are publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Our method is training-free, so no optimizer or training details apply. All test-time hyperparameters (particle counts, resampling rates, gradient scales, task parameters) are specified in Section 5 and Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Tables 9 and 10 report standard deviations across the evaluation set, as does Table 3 in the main paper. The 2D toy experiment in Appendix C.13 averages over 16,000 trials.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 5 states that experiments are run on NVIDIA A100 GPUs. Appendix C.10 (Table 16) reports wall-clock runtimes for each baseline sampler and each search method across different particle counts.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and our work conforms to it. We use only publicly available datasets and pretrained models.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss both positive impacts (improved scientific imaging, e.g., MRI) and negative impacts (privacy concerns from higher-fidelity identity reconstruction) at the end of Appendix E.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We do not release any new pretrained generative models or datasets. Our contribution is an inference-time algorithm built on top of existing publicly released models.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the original papers for all models (DPS, BlindDPS, DAPS, MPGD, ContextMRI, AdaFace, ImageReward, CLIP) and datasets (CelebA-HQ, FFHQ, ImageNet, fastMRI) used in our experiments, and respect their respective licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release our code with documentation and instructions for reproducing the main experimental results. The anonymized code is linked in the abstract.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: Our research does not involve crowdsourcing or human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: Our research does not involve human subjects and therefore does not require IRB approval.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: As described in Appendix F, we use ChatGPT to generate textual descriptions that serve as side information for a specific set of experiments, and we used GPT-5 for clarity and grammar improvements during writing. Neither use affects the core methodology.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.