Tempered Guided Diffusion

Andreas Makris
Department of Mathematics and Statistics
Lancaster University, UK
a.makris@lancaster.ac.uk

Paul Fearnhead
Department of Mathematics and Statistics
Lancaster University, UK
p.fearnhead@lancaster.ac.uk

Christopher Nemeth
Department of Mathematics and Statistics
Lancaster University, UK
c.nemeth@lancaster.ac.uk

Abstract

Training-free conditional diffusion provides a flexible alternative to task-specific conditional model training, but existing samplers often allocate computation inefficiently: independent guided trajectories can vary widely in quality, and additional function evaluations along a single trajectory may not recover from poor early decisions. We propose **Tempered Guided Diffusion (TGD)**, an annealed sequential Monte Carlo framework for training-free conditional sampling with diffusion priors. TGD targets tempered posterior distributions over the clean signal, using noisy diffusion states only as auxiliary variables for proposing reconstructions and propagating particles. Particles are reweighted by incremental likelihood ratios, resampled, and propagated across noise levels, concentrating computation on trajectories plausible under both the prior and observation. Under idealized exact-reconstruction assumptions, full TGD yields a consistent particle approximation to the posterior as the number of particles grows. For expensive reconstruction tasks, **Accelerated TGD (A-TGD)** retains early particle exploration but prunes to a single high-likelihood trajectory partway through sampling. TGD improves posterior approximation in controlled inverse problems with known posteriors, while A-TGD achieves favorable wall-clock speed-quality tradeoffs for single reconstructions on image inverse problems relative to independent multi-trajectory baselines. [1]

1 Introduction

Diffusion models [22, 24, 37, 42, 43, 45–47] have emerged as powerful generative priors for conditional generation [3, 12, 18, 21, 25, 38] and inverse problems [1, 7, 8, 13, 26, 27, 30, 31, 44, 48, 52, 55, 57]. In many applications, however, the conditioning mechanism changes from task to task, while paired training data for each new setting may be unavailable or costly to obtain. This makes training-free conditional diffusion [3, 20, 39, 49, 50, 53] especially attractive: rather than training a separate conditional model for every inverse problem, one reuses a pretrained unconditional diffusion prior and incorporates the observation only at sampling time. This flexibility is useful both when the goal is posterior sampling or to obtain a single high-quality reconstruction.

Despite this appeal, existing training-free conditional diffusion methods can be unreliable on challenging inverse problems. For a fixed observation, different random initializations may produce markedly different reconstructions, some of which remain inconsistent with the observation or have poor perceptual quality [55]. A common response is to run the sampler multiple times and retain the

[1] Code is available at https://github.com/zedt123/tgd_2026.

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

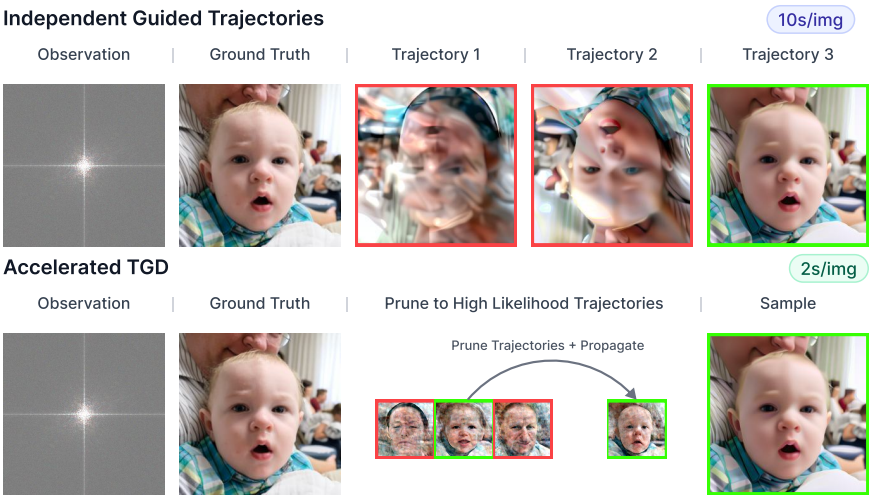


Figure 1: **Motivation for A-TGD.** Independent trajectories select only after completion; A-TGD explores early, prunes by likelihood, and completes one trajectory.

best output according to a data-consistency score [8, 55]. Such best-of-N heuristics can improve robustness, but they treat trajectories independently and spend computation uniformly across candidates, even after some have become unlikely to satisfy the observation. This suggests that the main difficulty is not only the amount of computation, but how it is allocated across candidate trajectories. This perspective is consistent with a broader theme in scalable Monte Carlo methodology [15]: when a single large computation is inefficient, one can often gain robustness by splitting computation across multiple partial simulations and then using weighting, selection, transformation, or recombination to concentrate effort on the statistically relevant components [40, 51]. In TGD, this principle is instantiated through interacting particles rather than through independent best-of-N trajectories.

Motivated by this perspective, we introduce *Tempered Guided Diffusion* (TGD), a sequential Monte Carlo (SMC) approach that maintains a population of candidate reconstructions and uses resampling to focus subsequent computation on the most promising candidates. Existing diffusion-SMC methods have applied particle-based ideas to conditional sampling and inverse problems [13, 49, 50]. TGD differs in where the annealed targets are defined: rather than treating noisy diffusion states as the main objects of inference, TGD defines a sequence of tempered posterior targets [10, 33–35] over the clean signal. For an inverse problem with clean signal $\boldsymbol{x}_0 \in \mathbb{R}^{d_x}$, observation $\boldsymbol{y} \in \mathbb{R}^{d_y}$, and likelihood $p(\boldsymbol{y} \mid \boldsymbol{x}_0)$, these targets are

$$\pi_r(\boldsymbol{x}_0) \propto p(\boldsymbol{x}_0)\, p(\boldsymbol{y} \mid \boldsymbol{x}_0)^{\lambda_r}, \qquad 0 \le \lambda_R \le \lambda_{R-1} \le \cdots \le \lambda_0 = 1, \tag{1}$$

where R is the number of restarts, $\lambda_r \in [0, 1]$ controls the strength of the observation, the index r decreases as conditioning increases, and $\pi_0 = p(\boldsymbol{x}_0 \mid \boldsymbol{y})$ is the full posterior. Noisy diffusion states are used only as auxiliary variables for reconstruction and propagation. At each stage, a training-free diffusion solver, such as an MPGD- or DAPS-style reconstruction module [20, 55], maps each noisy auxiliary particle to a candidate clean reconstruction. The candidates are then reweighted by the incremental likelihood ratio needed to move from π_r to π_{r-1}; resampling discards low-likelihood candidates and duplicates promising ones before the retained particles are re-noised for the next stage. Thus, unlike independent best-of-N sampling, TGD uses multiple reconstructions as an interacting population, concentrating later computation on candidates that remain plausible under both the prior and the observation.

For expensive image inverse problems, we use *Accelerated TGD* (A-TGD), which prunes the particle population once at an intermediate stage (see Figure 1) and completes the same schedule with a single high-likelihood trajectory.

Our contributions: (i) We introduce TGD, an annealed SMC framework for training-free conditional sampling over clean signals, and show that several existing training-free samplers can be viewed as different stagewise reconstruction modules within the same outer particle scheme. (ii) Under exact stagewise reconstruction and standard SMC assumptions, we show that idealized TGD

is asymptotically consistent for the posterior as the number of particles grows. (iii) We develop Accelerated TGD (A-TGD), a pruning-based variant that preserves early particle exploration while reducing the cost of reconstruction. (iv) We evaluate TGD in controlled inverse problems with known posteriors and on image inverse problems including inpainting and phase retrieval, showing improved posterior approximation in the controlled settings and that A-TGD improves wall-clock speed-quality tradeoffs for image reconstruction.

2 Preliminaries

Inverse problems. We consider inverse problems in which an unknown clean signal $\boldsymbol{x}_0 \in \mathbb{R}^{d_x}$ is observed indirectly through a measurement $\boldsymbol{y} \in \mathbb{R}^{d_y}$. The likelihood $p(\boldsymbol{y} \mid \boldsymbol{x}_0)$ encodes the probabilistic relationship between the clean signal and the observation, including the measurement process, or forward operator, and observation noise. Given a prior $p(\boldsymbol{x}_0)$ over clean signals, the posterior [15, 19, 32] is

$$p(\boldsymbol{x}_0 \mid \boldsymbol{y}) \propto p(\boldsymbol{x}_0)\, p(\boldsymbol{y} \mid \boldsymbol{x}_0). \tag{2}$$

We represent the prior $p(\boldsymbol{x}_0)$ using a pretrained diffusion model. We adopt a standard continuous-time formulation based on a forward noising process and its corresponding reverse dynamics.

Unconditional diffusion. Let $(\boldsymbol{x}_s)_{s\in[0,S]}$ denote a forward diffusion process defined by

$$\mathrm{d}\boldsymbol{x}_s = g(s)\,\mathrm{d}\boldsymbol{w}_s, \qquad \boldsymbol{x}_0 \sim p(\boldsymbol{x}_0),$$

where $\boldsymbol{w}_s$ is standard Brownian motion and $g(s)$ is a scalar noise schedule. This defines a family of marginal densities $p_s(\boldsymbol{x}_s)$, where p_s is the distribution obtained by sampling $\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0)$ and evolving the forward process to time s. For sufficiently large S, p_S is well-approximated by a simple Gaussian distribution [24, 47].

Under standard regularity conditions [2, 47], the reverse-time dynamics that transform $\boldsymbol{x}_S \sim p_S$ back to $\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0)$ are given by the reverse SDE

$$\mathrm{d}\boldsymbol{x}_s = \left[-g(s)^2 \nabla_{\boldsymbol{x}_s} \log p_s(\boldsymbol{x}_s)\right] \mathrm{d}s + g(s)\mathrm{d}\bar{\boldsymbol{w}}_s, \tag{3}$$

where $\bar{\boldsymbol{w}}_s$ denotes reverse-time Brownian motion. One can also use the corresponding probability-flow ODE [47],

$$\mathrm{d}\boldsymbol{x}_s = -\frac{1}{2} g(s)^2 \nabla_{\boldsymbol{x}_s} \log p_s(\boldsymbol{x}_s)\,\mathrm{d}s, \tag{4}$$

which has the same time-marginal distributions as the reverse SDE when integrated from S to 0. In practice, the score function $\nabla_{\boldsymbol{x}_s} \log p_s(\boldsymbol{x}_s)$ is approximated by a neural network trained on noisy samples [24, 47]. In the *Elucidated Diffusion Model* (EDM) parameterization [24], this corresponds to predicting a denoised estimate $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_s, s)$, yielding

$$\nabla_{\boldsymbol{x}_s} \log p_s(\boldsymbol{x}_s) = \frac{\hat{\boldsymbol{x}}_0(\boldsymbol{x}_s, s) - \boldsymbol{x}_s}{\sigma_s^2},$$

where $\sigma_s^2 = \int_0^s g(t)^2\,\mathrm{d}t$ is the accumulated noise variance of the forward process.

Training-free conditional sampling. Given an observation $\boldsymbol{y}$, we wish to sample from the posterior (2) without retraining the diffusion model. Training-free conditional diffusion uses the property that, if we adapt the reverse dynamics by replacing the unconditional score in either the reverse SDE (3) or the probability-flow ODE (4) with the conditional score, then simulating the resulting conditional dynamics to time 0 would yield samples from the posterior (2). The conditional score is defined using the conditional-score decomposition

$$\nabla_{\boldsymbol{x}_s} \log p_s(\boldsymbol{x}_s \mid \boldsymbol{y}) = \nabla_{\boldsymbol{x}_s} \log p_s(\boldsymbol{x}_s) + \nabla_{\boldsymbol{x}_s} \log p_s(\boldsymbol{y} \mid \boldsymbol{x}_s), \tag{5}$$

where

$$p_s(\boldsymbol{y} \mid \boldsymbol{x}_s) = \int p(\boldsymbol{y} \mid \boldsymbol{x}_0)\, p(\boldsymbol{x}_0 \mid \boldsymbol{x}_s)\,\mathrm{d}\boldsymbol{x}_0$$

is the observation likelihood marginalized through the diffusion conditional $p(\boldsymbol{x}_0 \mid \boldsymbol{x}_s)$.

Incorporating (5) into the reverse SDE yields condition-aware dynamics that map a noisy state $\boldsymbol{x}_s$ toward samples consistent with $\boldsymbol{y}$. In practice, however, the term $\nabla_{\boldsymbol{x}_s} \log p_s(\boldsymbol{y} \mid \boldsymbol{x}_s)$ is intractable and

must be approximated. Existing methods differ in this approximation: manifold-preserving guided diffusion (MPGD) [20] and related approaches [8, 39, 55] construct approximate gradients in data or measurement space, while others rely on plug-in likelihood approximations or score corrections.

Another common strategy is to introduce *restarts*, repeatedly reinitializing the diffusion process at intermediate noise levels and reapplying conditional updates [1, 3, 29, 53–55]. DAPS, for example, alternates noising and denoising steps and can use a coarse ODE discretization to obtain efficient reconstructions between restart levels [55]. Restarts can improve robustness, but they are typically used as an ad hoc sampling heuristic. In TGD, these short conditional reconstruction procedures are used as modules that map noisy auxiliary states to candidate clean signals; the outer procedure is instead formulated separately as SMC over an annealed sequence of clean-space posteriors.

3 Tempered Guided Diffusion

3.1 Annealed posterior paths over clean signals

Our goal is to sample from the posterior $p(\boldsymbol{x}_0 \mid \boldsymbol{y})$. Rather than imposing the full observation likelihood at once, TGD introduces an annealed sequence [5, 10, 33] of tempered targets [34, 36] over the clean signal $\boldsymbol{x}_0$,

$$\pi_r(\boldsymbol{x}_0) \propto p(\boldsymbol{x}_0)\, p(\boldsymbol{y} \mid \boldsymbol{x}_0)^{\lambda_r}, \qquad 0 \le \lambda_R \le \lambda_{R-1} \le \cdots \le \lambda_0 = 1, \tag{6}$$

where R is the number of restarts, $\lambda_r \in [0, 1]$ is the tempering parameter at stage r. The index r decreases as conditioning increases: $\pi_0 = p(\boldsymbol{x}_0 \mid \boldsymbol{y})$ is the full posterior, while earlier stages impose the observation more weakly. When $\lambda_R = 0$, the initial target is the prior. Practical choices of the tempering schedule are described in Appendix F.7.

The key modeling choice in TGD is that the annealed path (6) is defined over clean signals rather than noisy diffusion states. We refer to each π_r as a clean-space target because it is a distribution over the unknown clean signal $\boldsymbol{x}_0$. An outer stage pairs a tempering level λ_r with a diffusion noise level s_r. At stage r, the particle system is represented by noisy auxiliary states $\{\boldsymbol{z}_r^{(i)}\}_{i=1}^N$ at noise level s_r. These states are not themselves the inferential targets; they are diffusion-compatible variables used to propose and propagate clean candidates.

Given a noisy auxiliary state $\boldsymbol{z}_r^{(i)}$, a reconstruction step produces a candidate clean signal $\boldsymbol{x}_0^{(i)}$. These clean candidates are reweighted by the incremental likelihood ratio needed to move from π_r to π_{r-1} and may then be resampled. The retained clean particles are subsequently re-noised to form auxiliary states at the next noise level. Thus, the sequence $\{\pi_r\}_{r=0}^R$ defines the clean-space inference path, while diffusion supplies the reconstruction and re-noising operations used to move the particle population along that path.

We write $p_s(\boldsymbol{z} \mid \boldsymbol{x}_0)$ for the forward noising kernel and $p_s(\boldsymbol{x}_0 \mid \boldsymbol{z}) \propto p(\boldsymbol{x}_0) p_s(\boldsymbol{z} \mid \boldsymbol{x}_0)$ for the corresponding conditional density.

3.2 Stagewise construction of TGD

We approximate the targets $\{\pi_r\}_{r=0}^R$ with a weighted particle system evolved by SMC [6, 9, 14, 33]. At outer stage r, particles are represented by auxiliary states $\{(\boldsymbol{z}_r^{(i)}, w_r^{(i)})\}_{i=1}^N$ at noise level s_r, where

$$S = s_R \ge s_{R-1} \ge \cdots \ge s_0 > 0.$$

We initialize from the noisy prior,

$$\boldsymbol{z}_R^{(i)} \sim p_{s_R}(\boldsymbol{z}_R), \qquad w_R^{(i)} = \frac{1}{N}, \qquad i = 1, \ldots, N.$$

When $\lambda_R = 0$, this is exactly the noisy marginal corresponding to the initial clean-space target. More generally, it should be viewed as a prior initialization before the first tempered reconstruction step.

For $r = R, \ldots, 1$, one TGD update from stage r to stage $r - 1$ consists of: (i) reconstruction, (ii) weighting/resampling, and (iii) propagation.

Algorithm 1 Tempered Guided Diffusion (TGD)

Require: observation $\boldsymbol{y}$, number of particles N, tempering schedule $\{\lambda_r\}_{r=0}^{R}$, noise schedule $\{s_r\}_{r=0}^{R}$, conditional reconstruction law $p_{\lambda_r,s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r, \boldsymbol{y})$

Sample $\boldsymbol{z}_R^{(i)} \sim p_{s_R}(\boldsymbol{z}_R)$ and set $w_R^{(i)} \leftarrow 1/N$, for $i = 1, \dots, N$
for $r = R, R-1, \dots, 1$ **do**
  for $i = 1, \dots, N$ **do**
    Sample $\boldsymbol{x}_0^{(i)} \sim p_{\lambda_r,s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r^{(i)}, \boldsymbol{y})$
    $\tilde{w}_{r-1}^{(i)} \leftarrow w_r^{(i)}\, p(\boldsymbol{y} \mid \boldsymbol{x}_0^{(i)})^{\lambda_{r-1}-\lambda_r}$
  end for
  $w_{r-1}^{(i)} \leftarrow \tilde{w}_{r-1}^{(i)} / \sum_{j=1}^{N} \tilde{w}_{r-1}^{(j)}$, for $i = 1, \dots, N$
  Optionally resample according to $\{w_{r-1}^{(i)}\}_{i=1}^{N}$ and reset weights to $1/N$
  for $i = 1, \dots, N$ **do**
    Sample $\boldsymbol{z}_{r-1}^{(i)} \sim p_{s_{r-1}}(\boldsymbol{z}_{r-1} \mid \boldsymbol{x}_0^{(i)})$
  end for
end for
for $i = 1, \dots, N$ **do**
  Sample $\boldsymbol{x}_0^{(i)} \sim p_{\lambda_0,s_0}(\boldsymbol{x}_0 \mid \boldsymbol{z}_0^{(i)}, \boldsymbol{y})$
end for
return $\{(\boldsymbol{x}_0^{(i)}, w_0^{(i)})\}_{i=1}^{N}$

Reconstruction. Given $\boldsymbol{z}_r^{(i)}$, we sample a clean candidate from the stagewise reconstruction law

$$\boldsymbol{x}_0^{(i)} \sim p_{\lambda_r,s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r^{(i)}, \boldsymbol{y}),$$

where

$$p_{\lambda_r,s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r, \boldsymbol{y}) \propto p_{s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r)\, p(\boldsymbol{y} \mid \boldsymbol{x}_0)^{\lambda_r} \propto p(\boldsymbol{x}_0)\, p_{s_r}(\boldsymbol{z}_r \mid \boldsymbol{x}_0)\, p(\boldsymbol{y} \mid \boldsymbol{x}_0)^{\lambda_r}. \tag{7}$$

This maps each noisy auxiliary particle to a clean reconstruction under the current tempering level.

Weighting and resampling. To move from π_r to π_{r-1}, we update weights using the incremental likelihood ratio

$$\tilde{w}_{r-1}^{(i)} = w_r^{(i)}\, p(\boldsymbol{y} \mid \boldsymbol{x}_0^{(i)})^{\lambda_{r-1}-\lambda_r}.$$

After normalization, particles whose clean reconstructions better satisfy the observation receive larger weight as conditioning is strengthened. This likelihood-ratio form follows from the lifted construction underlying TGD (see Appendix A). To control weight degeneracy, one may resample particles using a standard low-variance scheme. In our experiments, when resampling is enabled, we use systematic resampling [33] and reset all weights to $1/N$.

Propagation. Each retained clean particle is propagated to the next outer stage by re-noising,

$$\boldsymbol{z}_{r-1}^{(i)} \sim p_{s_{r-1}}(\boldsymbol{z}_{r-1} \mid \boldsymbol{x}_0^{(i)}),$$

yielding a new population of noisy auxiliary states at level s_{r-1}.

After the outer loop, a final reconstruction at $r = 0$ produces clean samples

$$\boldsymbol{x}_0^{(i)} \sim p_{\lambda_0,s_0}(\boldsymbol{x}_0 \mid \boldsymbol{z}_0^{(i)}, \boldsymbol{y}),$$

and TGD returns the weighted particle set $\{(\boldsymbol{x}_0^{(i)}, w_0^{(i)})\}_{i=1}^{N}$.

3.3 Practical reconstruction modules

The ideal reconstruction law (7) is generally unavailable, so practical TGD replaces this draw with an approximate training-free conditional diffusion solver. We retain the ideal incremental likelihood potential, but the solver's induced reconstruction density is neither evaluated nor included in an importance correction. Practical TGD therefore approximates the ideal Feynman–Kac flow and is not

an exactly weighted SMC sampler for the stated targets. The exact weight derivation and consistency result apply only under exact stagewise reconstruction.

We instantiate this interface with MPGD-style and DAPS-style modules. MPGD/DPS-style methods approximate the conditional score term in (5), while DAPS-style methods combine unconditional diffusion reconstruction with clean-space observation correction. In all image experiments, each reconstruction module uses a coarse four-step ODE discretization between restart levels, so cost scales with the number of particles, outer stages, and reconstruction steps per stage.

This modular view treats existing training-free solvers as approximations to the same stagewise reconstruction law; detailed reductions are deferred to Appendices C and D.

3.4 Idealized consistency of TGD

We analyze an ideal reference version of TGD to isolate the particle approximation from reconstruction error. In this version, each reconstruction step samples exactly from

$$p_{\lambda_r,s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r, \boldsymbol{y}) \propto p_{s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r)\, p(\boldsymbol{y} \mid \boldsymbol{x}_0)^{\lambda_r},$$

and the full particle population is propagated without pruning. Exact reconstruction maps the noisy auxiliary marginal associated with π_r back to the clean target π_r; likelihood tempering then reweights particles from π_r to π_{r-1}, and re-noising produces the next auxiliary marginal. Thus ideal TGD is a lifted SMC sampler [10] over the clean-space targets.

Theorem 3.1 (Consistency of ideal TGD)**.** *Assume $\lambda_R = 0$, so that initialization from the noisy prior matches the initial clean-space target. Suppose further that the stagewise reconstruction kernel is exact at each outer stage, the incremental potentials*

$$G_r(\boldsymbol{x}_0) = p(\boldsymbol{y} \mid \boldsymbol{x}_0)^{\lambda_{r-1}-\lambda_r}$$

are bounded, the initial particles are sampled i.i.d. from the noisy prior, and resampling, when used, is performed with a standard SMC scheme [6, 33]. Then, for every bounded measurable test function h,

$$\sum_{i=1}^{N} w_0^{(i)} h(\boldsymbol{x}_0^{(i)}) \xrightarrow{\mathbb{P}} \mathbb{E}_{p(\boldsymbol{x}_0 \mid \boldsymbol{y})}[h(\boldsymbol{x}_0)].$$

Equivalently, $\hat{\pi}_0^N$ converges weakly in probability to the posterior $p(\boldsymbol{x}_0 \mid \boldsymbol{y})$.

The formal assumptions and proof are given in Appendix B. This result applies only to the ideal full-particle procedure; practical reconstruction modules and the pruning step in A-TGD introduce additional approximation.

3.5 Accelerated TGD

TGD is designed to approximate the posterior by propagating a particle population through all outer stages. For image inverse problems, where the goal is often a single reconstruction under a fixed compute budget, we use *Accelerated TGD* (A-TGD): a pruning-based variant that runs the full particle procedure only during the early stages.

A-TGD starts with N particles and follows TGD until a pruning fraction $\rho \in [0, 1]$ of the outer stages has been completed. At this point, it reconstructs clean candidates $\{\boldsymbol{x}_0^{(i)}\}_{i=1}^N$ using the current reconstruction module and retains the particle with largest likelihood,

$$i^\star = \arg\max_{i \in \{1,\dots,N\}} p(\boldsymbol{y} \mid \boldsymbol{x}_0^{(i)}).$$

All other particles are discarded, and the auxiliary state corresponding to the selected particle is continued through the remaining stages with a single trajectory. Thus A-TGD keeps the exploratory benefit of multiple early particles while avoiding the cost of propagating the full population to $r = 0$.

The pruning fraction ρ controls the trade-off between exploration and computation: smaller values prune earlier and are cheaper, while larger values more closely track full TGD. Because pruning collapses the particle population, A-TGD is a reconstruction-oriented acceleration and is not covered by the posterior-consistency result of Section 3.4.

4 Related Work

Training-free diffusion inverse solvers use a pretrained unconditional diffusion model and incorporate the observation only at sampling time. DPS [8], MPGD [20], DAPS [55], and related approaches [1, 39] differ in how they impose data consistency, for example through approximate conditional guidance, manifold constraints, decoupled posterior updates, or repeated corrections at intermediate noise levels. RePaint [29] and FreeDoM [54] further show that revisiting intermediate noise levels can improve robustness. TGD is complementary to these approaches: it treats conditional diffusion solvers as stagewise reconstruction modules inside an outer annealed particle procedure, rather than prescribing a single guidance approximation. The additional particle layer lets intermediate reconstructions interact through likelihood weighting and resampling, reallocating later computation toward candidates that better satisfy the observation. We make these links explicit in Appendix D, where several training-free samplers are recovered as special cases of the reconstruction interface.

TGD is also related to particle-based approaches for diffusion models and inverse problems, including SMCDiff [50], TDS [49], and FPS-SMC [13]. These methods demonstrate the usefulness of SMC ideas for diffusion-based sampling. More broadly, particle systems have also been used outside diffusion models for latent-variable inference and learning, including marginal maximum-likelihood training methods in which particles approximate intractable posterior or marginal quantities [28, 41]. TGD differs in where the annealed targets and particle weights are defined. Its SMC targets are clean-space distributions over $\boldsymbol{x}_0$, so each reconstructed candidate is weighted using the observation likelihood $p(\boldsymbol{y} \mid \boldsymbol{x}_0)$ evaluated on a clean signal. Noisy diffusion states are instead auxiliary variables: after resampling, retained clean particles are re-noised to produce the next auxiliary states and then reconstructed again. Thus, diffusion supplies proposal and refresh moves, while likelihood weighting and resampling act directly on clean reconstructions. This also distinguishes TGD from independent best-of-N sampling [55], where trajectories are run separately and selected only at the end; A-TGD uses this interaction early, then prunes for efficient reconstruction.

5 Experiments

Setup. We evaluate TGD in controlled inverse problems with two- and twenty-dimensional clean signals and known posteriors, and A-TGD on image inverse problems. The controlled problems use noisy elementwise absolute-value observations and measure posterior approximation using max-sliced Wasserstein distance (SWD) [4, 16, 17] to reference posterior samples. For images, we consider inpainting and phase retrieval on FFHQ [23] and ImageNet [11], reporting LPIPS [56], PSNR, and SSIM averaged over 100 validation images. Note that these image metrics assess single-reconstruction fidelity; they do not establish posterior coverage or calibration. In all image-result tables, entries are reported as mean $\pm$ standard deviation over the 100 validation images. We compare against DPS, DAPS with one trajectory, and DAPS with four independent trajectories, denoted DAPS $(N = 4)$, where the final reconstruction is selected by lowest measurement error. A-TGD also starts with $N = 4$ particles but prunes to a single trajectory partway through sampling. Image methods are compared at matched wall-clock runtime within each task and dataset. All image experiments use four Euler ODE steps per reconstruction module (except DPS, see Appendix F) and a single NVIDIA H200 GPU with 141GB memory.

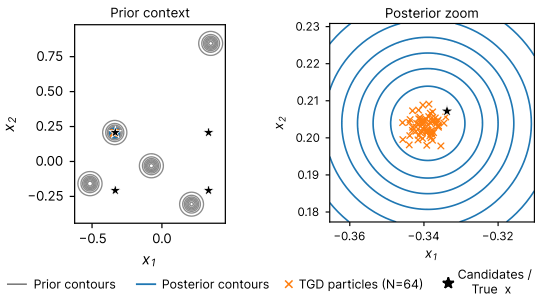
5.1 Controlled posterior approximation

We first evaluate whether the outer particle procedure improves posterior approximation in a controlled setting where the posterior is known. We consider a two-dimensional inverse problem with a Gaussian-mixture prior and noisy elementwise absolute-value observations,

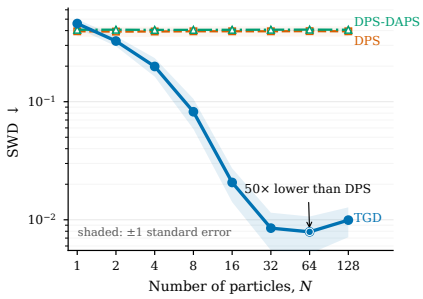
$$\boldsymbol{y} = |\boldsymbol{x}_0| + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \tau^2 I).$$

To isolate the particle inference mechanism from score-model error, we use the ground-truth unconditional score and compare final particles to reference posterior samples using SWD.

Figure 2 compares TGD to baselines and ablations as the number of particles increases. To make the comparison focus on the outer inference procedure, all methods use the DPS reconstruction proposal. TGD combines likelihood tempering with resampling, while the DPS-DAPS ablation removes these outer particle updates. TGD is the only method whose SWD consistently decreases toward zero as



(a) Prior context and posterior zoom.



(b) Posterior approximation vs. particle count.

Figure 2: **Controlled 2D inverse problem. Left:** prior context and posterior zoom for a representative test condition. Orange crosses denote the final TGD particles with $N = 64$; black markers denote the ground-truth clean sample and the sign-ambiguous candidates induced by the absolute-value observation. **Right:** SWD as a function of particle count. Bands denote ± 1 standard error over 10 test conditions. TGD improves rapidly with particle count, while DPS and DPS-DAPS do not benefit from additional independent particles in this setting. Lower is better.

the number of particles grows, empirically matching the behavior predicted by the idealized particle construction.

We also extend this inverse problem to twenty dimensions, with $x_0 \in \mathbb{R}^{20}$, and replace the DPS reconstruction proposal with exact draws from (7). This directly evaluates the ideal procedure in Theorem 3.1. With $N = 128$ particles, Exact-TGD achieves an SWD of 1.74×10^{-4} using 1,000 fixed projections, compared with 1.55×10^{-4} between independent exact-posterior sample sets. Appendix F.1 provides the experimental details, marginal-error measures, and effective-sample-size diagnostics.

5.2 Image inverse problems at matched runtime

Table 1 reports reconstruction quality at matched wall-clock runtime. Across both inpainting datasets, A-TGD achieves the best LPIPS and SSIM, while remaining competitive in PSNR. On FFHQ phase retrieval, where independent guided trajectories vary substantially in quality, A-TGD improves over DAPS $(N = 4)$ across all metrics. On ImageNet phase retrieval, the gains are more mixed: A-TGD matches DAPS $(N = 4)$ in PSNR, improves SSIM slightly, but does not improve LPIPS at the matched budget. Overall, these results suggest that A-TGD can use a fixed wall-clock budget more effectively than independent best-of-N sampling, especially when early multi-trajectory exploration identifies a high-likelihood trajectory that can be continued alone. Paired significance tests and ablations on particle count, pruning, annealing, resampling, and reconstruction modules are provided in Appendix G.

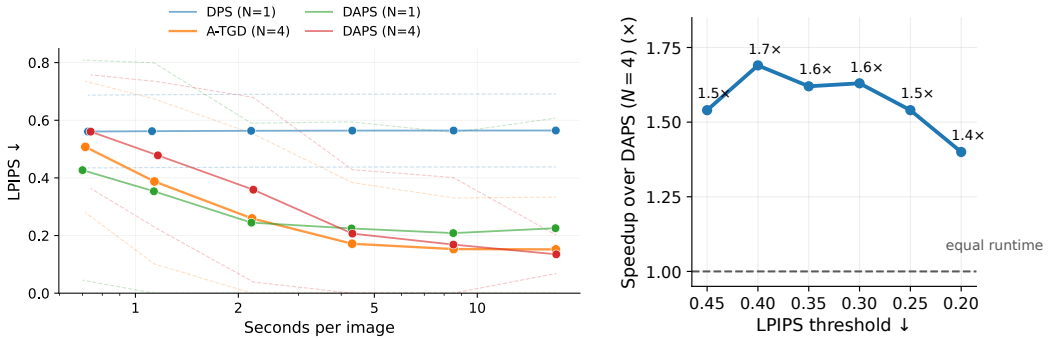
5.3 Speed-quality tradeoffs

The matched-runtime results in Table 1 compare methods at a fixed budget. We further evaluate how reconstruction quality changes with wall-clock time on FFHQ phase retrieval, where independent trajectories exhibit high variability and best-of-N selection is a strong baseline. Figure 3 plots LPIPS as a function of seconds per image for DPS, DAPS $(N = 1)$, DAPS $(N = 4)$, and A-TGD. DPS remains poor even as the number of function evaluations increases, while DAPS generally improves with runtime, especially for the best-of-four setting, but it must pay the cost of all independent trajectories throughout sampling. A-TGD achieves a better speed-quality curve by using multiple trajectories early and then pruning to a single high-likelihood trajectory for the remaining stages.

Figure 3(b) summarizes this tradeoff by reporting the speedup of A-TGD over DAPS $(N = 4)$ at fixed LPIPS thresholds. A-TGD reaches the same perceptual-quality thresholds faster across the evaluated range, suggesting that pruning retains much of the benefit of early multi-trajectory exploration while avoiding the cost of propagating all trajectories to completion.

Table 1: Image inverse-problem results at matched wall-clock runtime. DAPS ($N = 4$) runs four independent trajectories and selects by measurement error; A-TGD starts with four particles and prunes to one trajectory.

Task	Dataset	Algorithm	PSNR ↑	SSIM ↑	LPIPS ↓
Inpainting	FFHQ	A-TGD (ours)	$\mathbf{25.14}_{\pm 2.54}$	$\mathbf{0.853}_{\pm 0.025}$	$\mathbf{0.122}_{\pm 0.028}$
		DAPS ($N = 1$)	$24.93_{\pm 2.57}$	$0.801_{\pm 0.028}$	$0.153_{\pm 0.033}$
		DAPS ($N = 4$)	$24.52_{\pm 2.25}$	$0.790_{\pm 0.030}$	$0.157_{\pm 0.034}$
		DPS	$22.12_{\pm 2.51}$	$0.806_{\pm 0.030}$	$0.167_{\pm 0.033}$
	ImageNet	A-TGD (ours)	$\mathbf{20.89}_{\pm 3.67}$	$\mathbf{0.781}_{\pm 0.037}$	$\mathbf{0.204}_{\pm 0.036}$
		DAPS ($N = 1$)	$20.87_{\pm 3.65}$	$0.753_{\pm 0.035}$	$0.218_{\pm 0.044}$
		DAPS ($N = 4$)	$20.70_{\pm 3.42}$	$0.740_{\pm 0.033}$	$0.227_{\pm 0.044}$
		DPS	$18.39_{\pm 2.57}$	$0.710_{\pm 0.069}$	$0.281_{\pm 0.069}$
Phase Retrieval	FFHQ	A-TGD (ours)	$\mathbf{29.43}_{\pm 4.07}$	$\mathbf{0.811}_{\pm 0.104}$	$\mathbf{0.172}_{\pm 0.106}$
		DAPS ($N = 1$)	$26.98_{\pm 8.35}$	$0.767_{\pm 0.193}$	$0.225_{\pm 0.185}$
		DAPS ($N = 4$)	$27.81_{\pm 5.12}$	$0.765_{\pm 0.128}$	$0.207_{\pm 0.111}$
		DPS	$12.65_{\pm 2.42}$	$0.331_{\pm 0.095}$	$0.564_{\pm 0.063}$
	ImageNet	A-TGD (ours)	$22.18_{\pm 8.75}$	$\mathbf{0.617}_{\pm 0.276}$	$0.326_{\pm 0.218}$
		DAPS ($N = 1$)	$19.42_{\pm 9.31}$	$0.499_{\pm 0.298}$	$0.413_{\pm 0.222}$
		DAPS ($N = 4$)	$\mathbf{22.41}_{\pm 9.26}$	$0.593_{\pm 0.291}$	$\mathbf{0.325}_{\pm 0.221}$
		DPS	$12.58_{\pm 2.48}$	$0.232_{\pm 0.140}$	$0.601_{\pm 0.064}$



(a) LPIPS vs. wall-clock time

(b) Speedup to LPIPS threshold

Figure 3: **Speed-quality tradeoff on FFHQ phase retrieval. (a)** LPIPS as a function of wall-clock time per image for DPS, DAPS ($N = 1$), DAPS ($N = 4$), and A-TGD. **(b)** Speedup of A-TGD over DAPS ($N = 4$) to reach fixed LPIPS thresholds.

Figure 4 shows representative FFHQ reconstructions for inpainting and phase retrieval. The examples are consistent with Table 1: A-TGD preserves perceptual quality while improving observation consistency relative to independent trajectory baselines. In phase retrieval, DPS often fails to recover a plausible image, while DAPS improves substantially and A-TGD further benefits from early multi-trajectory exploration followed by pruning.

5.4 Oracle trajectory selection

Table 2 compares measurement-based selection with oracle selection among four independent DAPS trajectories. On ImageNet phase retrieval, oracle selection substantially improves DAPS, showing that measurement error can fail to identify the best available reconstruction. This effect is task-dependent: on FFHQ phase retrieval, A-TGD outperforms even oracle-selected DAPS, while on FFHQ inpainting it retains better LPIPS and SSIM but lower PSNR. These comparisons diagnose selection among DAPS trajectories; they do not directly quantify the effect of pruning in A-TGD. Since ground-truth images are unavailable at test time, developing better observation-based criteria for both particle pruning and terminal trajectory selection remains a direction for future work.

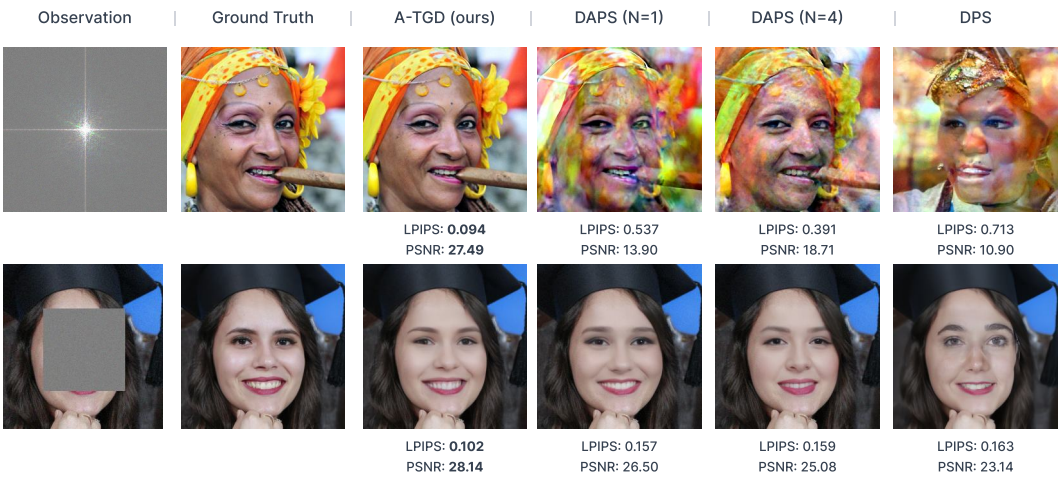


Figure 4: **Qualitative reconstructions on FFHQ.** We show one inpainting example and one phase retrieval example. A-TGD starts with four particles and prunes to one trajectory, while DAPS ($N = 4$) runs four independent trajectories and selects the final reconstruction with lowest measurement error.

Table 2: Trajectory-selection comparisons (means over 100 validation images per setting). DAPS-4 selects by lowest measurement error or the ground-truth metric indicated in parentheses; all three metrics are reported for the selected reconstruction.

Method / selection	PSNR ↑	LPIPS ↓	SSIM ↑
FFHQ inpainting			
A-TGD	25.14	0.122	0.853
DAPS-4 (measurement)	24.52	0.157	0.790
DAPS-4 (oracle LPIPS)	25.36	0.150	0.799
DAPS-4 (oracle PSNR)	25.65	0.152	0.799
DAPS-4 (oracle SSIM)	25.47	0.151	0.801
FFHQ phase retrieval			
A-TGD	29.43	0.172	0.811
DAPS-4 (measurement)	27.81	0.207	0.765
DAPS-4 (oracle LPIPS)	27.88	0.202	0.767
DAPS-4 (oracle PSNR)	28.00	0.204	0.767
DAPS-4 (oracle SSIM)	27.94	0.203	0.768
ImageNet phase retrieval			
A-TGD	22.18	0.326	0.617
DAPS-4 (measurement)	22.41	0.325	0.593
DAPS-4 (oracle LPIPS)	25.82	0.249	0.687
DAPS-4 (oracle PSNR)	25.95	0.250	0.689
DAPS-4 (oracle SSIM)	25.94	0.250	0.689

6 Discussion and Limitations

TGD separates clean-space particle inference from the conditional reconstruction module, yielding an idealized SMC construction while allowing practical implementations to reuse existing training-free diffusion solvers. The theoretical guarantee assumes exact stagewise reconstruction and an unpruned particle procedure; image experiments instead use approximate reconstruction modules, learned unconditional scores, and A-TGD pruning to obtain a single reconstruction under a fixed compute budget. The usefulness of particle-based exploration depends on the inverse problem: it is most effective when early trajectories reach different plausible reconstruction basins and the likelihood reliably selects among them. When the image prior and observation effectively determine a single solution, aggressive early resampling or pruning can discard a good trajectory and increase variance, especially in challenging phase retrieval settings.

Acknowledgments and Disclosure of Funding

The authors are grateful for the support of the Engineering and Physical Sciences Research Council (EPSRC) through the ProbAI Hub (grant EP/Y028783/1). This work was supported by Research England under the Expanding Excellence in England (E3) funding stream, which was awarded to MARS: Mathematics for AI in Real-world Systems in the School of Mathematical Sciences at Lancaster University. Authors declare no competing interests.

References

[1] Bilal Ahmed and Joseph G. Makin. Solving diffusion inverse problems with restart posterior sampling, 2025. URL https://arxiv.org/abs/2511.20705.

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

[3] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Roni Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=pzpWBbnwiJ.

[4] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51: 22–45, 2015. doi: 10.1007/s10851-014-0506-3. URL https://doi.org/10.1007/s10851-014-0506-3.

[5] Alberto Cabezas, Louis Sharrock, and Christopher Nemeth. Markovian flow matching: Accelerating mcmc with continuous normalizing flows. *Advances in Neural Information Processing Systems*, 37:104383–104411, 2024.

[6] Nicolas Chopin and Omiros Papaspiliopoulos. *An Introduction to Sequential Monte Carlo*. Springer Series in Statistics. Springer, Cham, 2020. doi: 10.1007/978-3-030-47845-2.

[7] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=nJJjv0JDJju.

[8] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

[9] Pierre Del Moral. *Feynman-Kac Formulae*, pages 47–93. Springer New York, New York, NY, 2004. ISBN 978-1-4684-9393-1. doi: 10.1007/978-1-4684-9393-1_2. URL https://doi.org/10.1007/978-1-4684-9393-1_2.

[10] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[12] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=AAWuCvzaVt.

[13] Zehao Dou and Yang Song. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=tplXNcHZs1.

[14] Arnaud Doucet, Nando de Freitas, and Neil Gordon. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer, New York, 2001.

[15] Paul Fearnhead, Christopher Nemeth, Chris J. Oates, and Chris Sherlock. *Scalable Monte Carlo for Bayesian Learning*. Institute of Mathematical Statistics Monographs. Cambridge University Press, 2025.

[16] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. URL http://jmlr.org/papers/v22/20-451.html.

[17] Rémi Flamary, Cédric Vincent-Cuaz, Nicolas Courty, Alexandre Gramfort, Oleksii Kachaiev, Huy Quang Tran, Laurène David, Clément Bonet, Nathan Cassereau, Théo Gnassounou, Eloi Tanguy, Julie Delon, Antoine Collas, Sonia Mazelet, Laetitia Chapel, Tanguy Kerdoncuff, Xizheng Yu, Matthew Feickert, Paul Krzakala, Tianlin Liu, and Eduardo Fernandes Montesuma. Pot python optimal transport (version 0.9.5), 2024. URL https://github.com/PythonOT/POT.

[18] Zhengqi Gao, Kaiwen Zha, Tianyuan Zhang, Zihui Xue, and Duane S Boning. REG: Rectified gradient guidance for conditional diffusion models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=uNK4ftGdnq.

[19] Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

[20] Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J Zico Kolter, Ruslan Salakhutdinov, and Stefano Ermon. Manifold preserving guided diffusion. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=o3BxOLoxm1.

[21] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

[23] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019.

[24] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=k7FuTOWMOc7.

[25] Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=bg6fVPVs3s.

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

[27] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 23593–23606. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/95504595b6169131b6ed6cd72eb05616-Paper-Conference.pdf.

[28] Juan Kuntz, Jen Ning Lim, and Adam M Johansen. Particle algorithms for maximum likelihood training of latent variable models. In *International Conference on Artificial Intelligence and Statistics*, pages 5134–5180. PMLR, 2023.

[29] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[30] Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving inverse problems with diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=1YO4EE3SPB.

[31] Naoki Murata, Koichi Saito, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Yuki Mitsufuji, and Stefano Ermon. GibbsDDRM: A partially collapsed gibbs sampler for solving blind inverse problems with denoising diffusion restoration. In *International Conference on Machine Learning*, 2023.

[32] Kevin P Murphy. *Probabilistic machine learning: an introduction*. MIT press, 2022.

[33] Christian A. Naesseth, Fredrik Lindsten, and Thomas B. Schön. Elements of sequential monte carlo. *Found. Trends Mach. Learn.*, 12(3):307–392, November 2019. ISSN 1935-8237. doi: 10.1561/2200000074. URL https://doi.org/10.1561/2200000074.

[34] Radford M Neal. Sampling from multimodal distributions using tempered transitions. *Statistics and computing*, 6(4):353–366, 1996.

[35] Radford M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, April 2001. ISSN 1573-1375. doi: 10.1023/A:1008923215028. URL https://doi.org/10.1023/A:1008923215028.

[36] Christopher Nemeth, Fredrik Lindsten, Maurizio Filippone, and James Hensman. Pseudo-extended markov chain monte carlo. *Advances in Neural Information Processing Systems*, 32, 2019.

[37] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171. PMLR, 2021. URL https://proceedings.mlr.press/v139/nichol21a.html.

[38] Kushagra Pandey, Farrin Marouf Sofian, Felix Draxler, Theofanis Karaletsos, and Stephan Mandt. Variational control for guidance in diffusion models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Z0ffRRtOim.

[39] Severi Rissanen, Markus Heinonen, and Arno Solin. Free hunch: Denoiser covariance estimation for diffusion models without extra costs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4JK2XMGUc8.

[40] Steven L. Scott, Alexander W. Blocker, Fernando V. Bonassi, Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. Bayes and big data: the consensus monte carlo algorithm. *International Journal of Management Science and Engineering Management*, 11(2):78–88, 2016. doi: 10.1080/17509653.2016.1142191. URL https://doi.org/10.1080/17509653.2016.1142191.

[41] Louis Sharrock, Daniel Dodd, and Christopher Nemeth. Tuning-free maximum likelihood training of latent variable models via coin betting. In *International Conference on Artificial Intelligence and Statistics*, pages 1810–1818. PMLR, 2024.

[42] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

[43] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

[44] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9_gsMA8MRKQ.

[45] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/3001ef257407d5a371a96dcd947c7d93-Paper.pdf.

[46] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12438–12448. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/92c3b916311a5517d9290576e3ea37ad-Paper.pdf.

[47] 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*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

[48] Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=vaRCHVj0uGI.

[49] Brian L. Trippe, Luhuan Wu, Christian A. Naesseth, David Blei, and John Patrick Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023. URL https://openreview.net/forum?id=r9s3Gbxz7g.

[50] Brian L. Trippe, Jason Yim, Doug Tischer, David Baker, Tamara Broderick, Regina Barzilay, and Tommi S. Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6TxBxqNME1Y.

[51] Callum Vyner, Christopher Nemeth, and Chris Sherlock. Swiss: A scalable markov chain monte carlo divide-and-conquer strategy. *Stat*, 12(1):e523, 2023.

[52] Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=mRieQgMtNTQ.

[53] Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Zou, and Stefano Ermon. Tfg: Unified training-free guidance for diffusion models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 22370–22417. Curran Associates, Inc., 2024. doi: 10.52202/079017-0704. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/2818054fc6de6dacdda0f142a3475933-Paper-Conference.pdf.

[54] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[55] Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20895–20905. IEEE, June 2025. doi: 10.1109/cvpr52734.2025.01946. URL http://dx.doi.org/10.1109/CVPR52734.2025.01946.

[56] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.

[57] Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising diffusion models for plug-and-play image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (NTIRE)*, 2023.

A Lifted SMC Construction and Incremental Weights

We derive the incremental weight update used in the stagewise TGD construction. Throughout this appendix, densities are understood up to normalizing constants that do not depend on the particle index, since SMC weights are normalized after each outer stage. For compactness, write

$$\ell(\boldsymbol{x}_0) := p(\boldsymbol{y} \mid \boldsymbol{x}_0).$$

Recall that the clean-space annealed target at outer stage r is

$$\pi_r(\boldsymbol{x}_0) \propto p(\boldsymbol{x}_0)\ell(\boldsymbol{x}_0)^{\lambda_r}, \qquad 0 \le \lambda_R \le \cdots \le \lambda_0 = 1. \tag{8}$$

To connect this target to the diffusion prior, TGD introduces an auxiliary noisy state $\boldsymbol{z}_r$ at noise level s_r. Define the unnormalized lifted density

$$\gamma_r(\boldsymbol{x}_0, \boldsymbol{z}_r) := p(\boldsymbol{x}_0)\ell(\boldsymbol{x}_0)^{\lambda_r} p_{s_r}(\boldsymbol{z}_r \mid \boldsymbol{x}_0). \tag{9}$$

Its clean marginal is proportional to π_r, since $\int p_{s_r}(\boldsymbol{z}_r \mid \boldsymbol{x}_0)\, d\boldsymbol{z}_r = 1$. Its noisy marginal is

$$\eta_r(\boldsymbol{z}_r) := \int \gamma_r(\boldsymbol{x}_0, \boldsymbol{z}_r)\, d\boldsymbol{x}_0. \tag{10}$$

The ideal TGD reconstruction kernel at stage r is the conditional distribution induced by γ_r:

$$K_r(d\boldsymbol{x}_0 \mid \boldsymbol{z}_r) := p_{\lambda_r, s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r, \boldsymbol{y})\, d\boldsymbol{x}_0 = \frac{\gamma_r(\boldsymbol{x}_0, \boldsymbol{z}_r)}{\eta_r(\boldsymbol{z}_r)}\, d\boldsymbol{x}_0. \tag{11}$$

Equivalently,

$$p_{\lambda_r, s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r, \boldsymbol{y}) \propto p_{s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r)\ell(\boldsymbol{x}_0)^{\lambda_r} \propto p(\boldsymbol{x}_0) p_{s_r}(\boldsymbol{z}_r \mid \boldsymbol{x}_0)\ell(\boldsymbol{x}_0)^{\lambda_r}. \tag{12}$$

Suppose a particle $\boldsymbol{z}_r$ targets the noisy marginal η_r. One ideal TGD transition first samples

$$\boldsymbol{x}_0 \sim K_r(\cdot \mid \boldsymbol{z}_r),$$

and then re-noises the retained clean sample to the next noise level:

$$\boldsymbol{z}_{r-1} \sim p_{s_{r-1}}(\cdot \mid \boldsymbol{x}_0).$$

Thus the proposal kernel from $\boldsymbol{z}_r$ to $(\boldsymbol{x}_0, \boldsymbol{z}_{r-1})$ is

$$Q_r(d\boldsymbol{x}_0, d\boldsymbol{z}_{r-1} \mid \boldsymbol{z}_r) = K_r(d\boldsymbol{x}_0 \mid \boldsymbol{z}_r)\, p_{s_{r-1}}(d\boldsymbol{z}_{r-1} \mid \boldsymbol{x}_0). \tag{13}$$

To derive the incremental weight, define the following unnormalized extended target:

$$\Gamma_{r-1}(\boldsymbol{z}_r, \boldsymbol{x}_0, \boldsymbol{z}_{r-1}) := p(\boldsymbol{x}_0)\ell(\boldsymbol{x}_0)^{\lambda_{r-1}} p_{s_r}(\boldsymbol{z}_r \mid \boldsymbol{x}_0) p_{s_{r-1}}(\boldsymbol{z}_{r-1} \mid \boldsymbol{x}_0). \tag{14}$$

This is a valid extension of the next-stage lifted target because marginalizing out $\boldsymbol{z}_r$ gives

$$\begin{aligned}\int \Gamma_{r-1}(\boldsymbol{z}_r, \boldsymbol{x}_0, \boldsymbol{z}_{r-1})\, d\boldsymbol{z}_r &= p(\boldsymbol{x}_0)\ell(\boldsymbol{x}_0)^{\lambda_{r-1}} p_{s_{r-1}}(\boldsymbol{z}_{r-1} \mid \boldsymbol{x}_0) \int p_{s_r}(\boldsymbol{z}_r \mid \boldsymbol{x}_0)\, d\boldsymbol{z}_r \\ &= p(\boldsymbol{x}_0)\ell(\boldsymbol{x}_0)^{\lambda_{r-1}} p_{s_{r-1}}(\boldsymbol{z}_{r-1} \mid \boldsymbol{x}_0) = \gamma_{r-1}(\boldsymbol{x}_0, \boldsymbol{z}_{r-1}).\end{aligned} \tag{15}$$

The standard SMC incremental weight associated with the proposal $\eta_r(\boldsymbol{z}_r)Q_r(d\boldsymbol{x}_0, d\boldsymbol{z}_{r-1} \mid \boldsymbol{z}_r)$ and the extended target Γ_{r-1} is

$$G_r(\boldsymbol{z}_r, \boldsymbol{x}_0, \boldsymbol{z}_{r-1}) = \frac{\Gamma_{r-1}(\boldsymbol{z}_r, \boldsymbol{x}_0, \boldsymbol{z}_{r-1})}{\eta_r(\boldsymbol{z}_r) Q_r(\boldsymbol{x}_0, \boldsymbol{z}_{r-1} \mid \boldsymbol{z}_r)}, \tag{16}$$

up to a stage-dependent normalizing constant.

Using (11) and (13), the denominator becomes

$$\begin{aligned}\eta_r(\boldsymbol{z}_r) Q_r(\boldsymbol{x}_0, \boldsymbol{z}_{r-1} \mid \boldsymbol{z}_r) &= \eta_r(\boldsymbol{z}_r) \frac{\gamma_r(\boldsymbol{x}_0, \boldsymbol{z}_r)}{\eta_r(\boldsymbol{z}_r)} p_{s_{r-1}}(\boldsymbol{z}_{r-1} \mid \boldsymbol{x}_0) \\ &= \gamma_r(\boldsymbol{x}_0, \boldsymbol{z}_r) p_{s_{r-1}}(\boldsymbol{z}_{r-1} \mid \boldsymbol{x}_0) \\ &= p(\boldsymbol{x}_0)\ell(\boldsymbol{x}_0)^{\lambda_r} p_{s_r}(\boldsymbol{z}_r \mid \boldsymbol{x}_0) p_{s_{r-1}}(\boldsymbol{z}_{r-1} \mid \boldsymbol{x}_0).\end{aligned} \tag{17}$$

Comparing (17) with the extended target (14), all terms cancel except the change in likelihood tempering:

$$G_r(\boldsymbol{z}_r, \boldsymbol{x}_0, \boldsymbol{z}_{r-1}) \propto \ell(\boldsymbol{x}_0)^{\lambda_{r-1}-\lambda_r} = p(\boldsymbol{y} \mid \boldsymbol{x}_0)^{\lambda_{r-1}-\lambda_r}. \tag{18}$$

Therefore, for particle i,

$$\widetilde{w}_{r-1}^{(i)} = w_r^{(i)} p(\boldsymbol{y} \mid \boldsymbol{x}_0^{(i)})^{\lambda_{r-1}-\lambda_r}. \tag{19}$$

The incremental weight does not depend on the proposed auxiliary state $\boldsymbol{z}_{r-1}$. Consequently, the weighting and resampling step can be performed immediately after reconstruction, before re-noising to the next level. This gives exactly the stagewise update used by ideal TGD in Section 3.2.

B Consistency of Ideal TGD

We state and prove the asymptotic consistency result for ideal TGD: the version of TGD in which each stagewise reconstruction step samples exactly from the tempered conditional law introduced in Section 3.2. This result applies to the full particle procedure with exact reconstruction kernels. It does not apply to approximate reconstruction modules or to the pruning step used by A-TGD.

Throughout this appendix, all densities are taken with respect to fixed dominating measures, and we use the same notation for a density and its associated measure when the meaning is clear. For compactness, write

$$\ell(\boldsymbol{x}_0) := p(\boldsymbol{y} \mid \boldsymbol{x}_0).$$

For $r = 0, \dots, R$, define the unnormalized clean-space density

$$\gamma_r(\boldsymbol{x}_0) := p(\boldsymbol{x}_0)\ell(\boldsymbol{x}_0)^{\lambda_r}, \qquad Z_r := \int \gamma_r(\boldsymbol{u})\, d\boldsymbol{u}. \tag{20}$$

The corresponding normalized clean-space target is

$$\pi_r(d\boldsymbol{x}_0) := \frac{\gamma_r(\boldsymbol{x}_0)}{Z_r}\, d\boldsymbol{x}_0. \tag{21}$$

Since $\lambda_0 = 1$, π_0 is the posterior $p(\boldsymbol{x}_0 \mid \boldsymbol{y})$.

At outer stage r, TGD uses an auxiliary noisy state $\boldsymbol{z}_r$ at noise level s_r. The corresponding noisy auxiliary marginal is

$$\eta_r(\boldsymbol{z}_r) := \int \frac{\gamma_r(\boldsymbol{x}_0)}{Z_r} p_{s_r}(\boldsymbol{z}_r \mid \boldsymbol{x}_0)\, d\boldsymbol{x}_0, \qquad \eta_r(d\boldsymbol{z}_r) := \eta_r(\boldsymbol{z}_r)\, d\boldsymbol{z}_r. \tag{22}$$

Theorem B.1 (Consistency of ideal TGD)**.** *Assume the following conditions.*

1. ***Well-defined targets.*** *For every $r = 0, \dots, R$, $0 < Z_r < \infty$, and the tempering schedule satisfies*
$$0 \le \lambda_R \le \lambda_{R-1} \le \cdots \le \lambda_0 = 1.$$
2. ***Initialization.*** *$\lambda_R = 0$, and the initial particles are sampled i.i.d. from the noisy prior:*
$$\boldsymbol{z}_R^{(i)} \sim p_{s_R}(\boldsymbol{z}_R), \qquad w_R^{(i)} = 1/N.$$
Equivalently, the initial weighted empirical measure is consistent for η_R.
3. ***Exact reconstruction.*** *For each $r = 0, \dots, R$, the stagewise reconstruction kernel used by ideal TGD is the exact conditional law*
$$K_r(d\boldsymbol{x}_0 \mid \boldsymbol{z}_r) := p_{\lambda_r, s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r, \boldsymbol{y})\, d\boldsymbol{x}_0 = \frac{\pi_r(\boldsymbol{x}_0) p_{s_r}(\boldsymbol{z}_r \mid \boldsymbol{x}_0)}{\eta_r(\boldsymbol{z}_r)}\, d\boldsymbol{x}_0. \tag{23}$$
Equivalently,
$$p_{\lambda_r, s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r, \boldsymbol{y}) \propto p_{s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r)\ell(\boldsymbol{x}_0)^{\lambda_r}. \tag{24}$$
4. ***Bounded incremental potentials.*** *For each $r = R, \dots, 1$, the incremental potential*
$$G_r(\boldsymbol{x}_0) := \ell(\boldsymbol{x}_0)^{\lambda_{r-1}-\lambda_r} = p(\boldsymbol{y} \mid \boldsymbol{x}_0)^{\lambda_{r-1}-\lambda_r} \tag{25}$$
is measurable and bounded, with $\pi_r(G_r) > 0$.

5. ***Consistent resampling.*** *At each outer stage, either resampling is omitted or the resampling scheme preserves consistency of weighted empirical measures: whenever the weighted particle system is consistent for a target distribution, the equally weighted resampled particles are also consistent for the same target. Standard SMC schemes [6, 33] such as multinomial, stratified, residual, and systematic resampling satisfy this condition.*

Then, for every bounded measurable test function h, the final weighted empirical measure produced by ideal TGD satisfies

$$\sum_{i=1}^{N} w_0^{(i)} h(\boldsymbol{x}_0^{(i)}) \xrightarrow{\mathbb{P}} \int h(\boldsymbol{x}_0)\, \pi_0(d\boldsymbol{x}_0) = \mathbb{E}_{p(\boldsymbol{x}_0|\boldsymbol{y})}[h(\boldsymbol{x}_0)]. \tag{26}$$

In particular, the final weighted empirical measure produced by ideal TGD converges weakly in probability to the posterior $p(\boldsymbol{x}_0 \mid \boldsymbol{y})$ as $N \to \infty$.

Proof. We prove the result by identifying ideal TGD as a standard SMC approximation of an exact sequence of measure transformations.

Step 1: One ideal TGD stage has the correct population flow. Fix $r \in \{R, \dots, 1\}$. By the definition of the exact reconstruction kernel,

$$K_r(d\boldsymbol{x}_0 \mid \boldsymbol{z}_r) = \frac{\pi_r(\boldsymbol{x}_0) p_{s_r}(\boldsymbol{z}_r \mid \boldsymbol{x}_0)}{\eta_r(\boldsymbol{z}_r)}\, d\boldsymbol{x}_0. \tag{27}$$

Integrating this kernel against η_r gives

$$\begin{aligned} \eta_r K_r(d\boldsymbol{x}_0) &:= \int \eta_r(d\boldsymbol{z}_r) K_r(d\boldsymbol{x}_0 \mid \boldsymbol{z}_r) \\ &= \int \eta_r(\boldsymbol{z}_r)\, d\boldsymbol{z}_r \frac{\pi_r(\boldsymbol{x}_0) p_{s_r}(\boldsymbol{z}_r \mid \boldsymbol{x}_0)}{\eta_r(\boldsymbol{z}_r)}\, d\boldsymbol{x}_0 \\ &= \pi_r(\boldsymbol{x}_0) \left(\int p_{s_r}(\boldsymbol{z}_r \mid \boldsymbol{x}_0)\, d\boldsymbol{z}_r \right) d\boldsymbol{x}_0 = \pi_r(d\boldsymbol{x}_0). \end{aligned} \tag{28}$$

Thus exact reconstruction maps the noisy marginal η_r to the clean target π_r.

Next define the Boltzmann–Gibbs reweighting operator

$$\Psi_{G_r}(\mu)(d\boldsymbol{x}_0) := \frac{G_r(\boldsymbol{x}_0)\mu(d\boldsymbol{x}_0)}{\mu(G_r)}. \tag{29}$$

Applying this operator to π_r gives

$$\begin{aligned} \Psi_{G_r}(\pi_r)(d\boldsymbol{x}_0) &\propto p(\boldsymbol{y} \mid \boldsymbol{x}_0)^{\lambda_{r-1}-\lambda_r} p(\boldsymbol{x}_0) p(\boldsymbol{y} \mid \boldsymbol{x}_0)^{\lambda_r} d\boldsymbol{x}_0 \\ &= p(\boldsymbol{x}_0) p(\boldsymbol{y} \mid \boldsymbol{x}_0)^{\lambda_{r-1}} d\boldsymbol{x}_0 \propto \pi_{r-1}(d\boldsymbol{x}_0). \end{aligned} \tag{30}$$

Therefore,

$$\Psi_{G_r}(\pi_r) = \pi_{r-1}. \tag{31}$$

Finally, let

$$P_{r-1}(d\boldsymbol{z}_{r-1} \mid \boldsymbol{x}_0) := p_{s_{r-1}}(d\boldsymbol{z}_{r-1} \mid \boldsymbol{x}_0) \tag{32}$$

be the forward noising kernel to the next auxiliary noise level. By the definition of η_{r-1},

$$\pi_{r-1} P_{r-1} = \eta_{r-1}. \tag{33}$$

Combining (28), (31), and (33), one ideal TGD stage implements the exact population flow

$$\eta_r \xrightarrow{K_r} \pi_r \xrightarrow{\Psi_{G_r}} \pi_{r-1} \xrightarrow{P_{r-1}} \eta_{r-1}. \tag{34}$$

At the terminal stage, there is no additional reweighting or propagation; exact reconstruction gives

$$\eta_0 \xrightarrow{K_0} \pi_0. \tag{35}$$

Step 2: The particle update consistently approximates this flow. Assume that the weighted empirical measure

$$\widehat{\eta}_r^N := \sum_{i=1}^N w_r^{(i)} \delta_{\boldsymbol{z}_r^{(i)}} \tag{36}$$

is consistent for η_r, meaning that for every bounded measurable φ,

$$\widehat{\eta}_r^N(\varphi) \xrightarrow{\mathbb{P}} \eta_r(\varphi).$$

The ideal TGD update first samples

$$\boldsymbol{x}_0^{(i)} \sim K_r(\cdot \mid \boldsymbol{z}_r^{(i)}).$$

By standard consistency results for sequential Monte Carlo samplers [9, 14, 33], mutation by a Markov kernel preserves consistency. Hence the resulting weighted empirical measure over clean particles is consistent for

$$\eta_r K_r = \pi_r.$$

TGD then applies the incremental potential

$$G_r(\boldsymbol{x}_0) = p(\boldsymbol{y} \mid \boldsymbol{x}_0)^{\lambda_{r-1}-\lambda_r},$$

using the weight update

$$\widetilde{w}_{r-1}^{(i)} \propto w_r^{(i)} G_r(\boldsymbol{x}_0^{(i)}) = w_r^{(i)} p(\boldsymbol{y} \mid \boldsymbol{x}_0^{(i)})^{\lambda_{r-1}-\lambda_r}. \tag{37}$$

This is the same incremental weight derived in Appendix A. Since G_r is bounded and $\pi_r(G_r) > 0$, standard SMC consistency results imply that weighting preserves consistency, and the weighted clean empirical measure is consistent for

$$\Psi_{G_r}(\pi_r) = \pi_{r-1}.$$

If resampling is performed, the assumed consistency of the resampling scheme preserves this limit. If resampling is omitted, the weighted particle system is left unchanged and consistency is immediate.

Finally, TGD samples

$$\boldsymbol{z}_{r-1}^{(i)} \sim P_{r-1}(\cdot \mid \boldsymbol{x}_0^{(i)}).$$

A second application of mutation consistency gives a weighted empirical measure over noisy auxiliary particles that is consistent for

$$\pi_{r-1} P_{r-1} = \eta_{r-1}.$$

Thus, consistency for η_r implies consistency for η_{r-1}.

At the terminal stage $r = 0$, if

$$\widehat{\eta}_0^N = \sum_{i=1}^N w_0^{(i)} \delta_{\boldsymbol{z}_0^{(i)}}$$

is consistent for η_0, then sampling

$$\boldsymbol{x}_0^{(i)} \sim K_0(\cdot \mid \boldsymbol{z}_0^{(i)})$$

produces a weighted clean empirical measure consistent for

$$\eta_0 K_0 = \pi_0.$$

Step 3: Induction over outer stages. Because $\lambda_R = 0$, the initial clean target is the prior:

$$\pi_R(d\boldsymbol{x}_0) = p(\boldsymbol{x}_0) d\boldsymbol{x}_0.$$

Therefore the corresponding noisy marginal is

$$\eta_R(d\boldsymbol{z}_R) = \int p(\boldsymbol{x}_0) p_{s_R}(d\boldsymbol{z}_R \mid \boldsymbol{x}_0)\, d\boldsymbol{x}_0 = p_{s_R}(d\boldsymbol{z}_R),$$

which is exactly the noisy prior used to initialize TGD. Hence the initial empirical measure is consistent for η_R.

Applying Step 2 for

$$r = R, R-1, \dots, 1$$

shows by induction that the particle system is consistent for η_0. Applying the terminal reconstruction kernel K_0 then yields a weighted empirical measure over clean particles that is consistent for

$$\pi_0(d\boldsymbol{x}_0) = p(\boldsymbol{x}_0 \mid \boldsymbol{y}) d\boldsymbol{x}_0.$$

Consequently, for every bounded measurable h,

$$\sum_{i=1}^{N} w_0^{(i)} h(\boldsymbol{x}_0^{(i)}) \xrightarrow{\mathbb{P}} \pi_0(h) = \mathbb{E}_{p(\boldsymbol{x}_0 \mid \boldsymbol{y})}[h(\boldsymbol{x}_0)].$$

This proves the claim. □

C Hybrid Conditional Reconstruction and Clean-Space Correction

Section 3.3 described two practical families of reconstruction modules for approximating the ideal stagewise law

$$p_{\lambda_r, s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r, \boldsymbol{y}) \propto p_{s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r)\, p(\boldsymbol{y} \mid \boldsymbol{x}_0)^{\lambda_r}, \tag{38}$$

where $\boldsymbol{z}_r$ is the auxiliary state at noise level s_r. Conditional reverse-dynamics modules, such as DPS or MPGD solvers, attempt to impose the observation during the reconstruction dynamics. Clean-space correction modules, such as DAPS solvers, first reconstruct using the unconditional diffusion prior and then correct the clean sample using the observation. We show that these two strategies can be viewed as endpoints of a common hybrid construction.

For compactness, write

$$\ell(\boldsymbol{x}_0) := p(\boldsymbol{y} \mid \boldsymbol{x}_0).$$

For any intermediate conditioning level $\nu \in [0, \lambda_r]$, define the weaker stagewise conditional law

$$p_{\nu, s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r, \boldsymbol{y}) \propto p_{s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r)\, \ell(\boldsymbol{x}_0)^{\nu}. \tag{39}$$

Then the full stagewise law admits the factorization

$$\begin{aligned} p_{\lambda_r, s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r, \boldsymbol{y}) &\propto p_{s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r)\, \ell(\boldsymbol{x}_0)^{\lambda_r} \\ &= p_{s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r)\, \ell(\boldsymbol{x}_0)^{\nu} \ell(\boldsymbol{x}_0)^{\lambda_r - \nu} \\ &\propto p_{\nu, s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r, \boldsymbol{y})\, \ell(\boldsymbol{x}_0)^{\lambda_r - \nu}. \end{aligned} \tag{40}$$

Thus, one may first approximately sample from a weaker conditional law and then apply a clean-space correction that accounts for the remaining likelihood exponent.

This suggests the following hybrid reconstruction module at outer stage r.

Conditional reverse-dynamics warm start. Choose an intermediate conditioning level $\nu \in [0, \lambda_r]$. Starting from the auxiliary noisy state $\boldsymbol{z}_r$, run a conditional reverse-dynamics reconstruction module whose target is the weaker law

$$p_{\nu, s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r, \boldsymbol{y}) \propto p_{s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r)\, \ell(\boldsymbol{x}_0)^{\nu}.$$

Let the resulting clean candidate be denoted by $\widetilde{\boldsymbol{x}}_0$. In practice, this step is approximate, because conditional reverse dynamics use tractable approximations to the intractable conditional score.

Clean-space correction. Starting from $\widetilde{\boldsymbol{x}}_0$, apply a clean-space Markov kernel $M_{r,\nu}$ whose invariant distribution is the full stagewise law

$$p_{\lambda_r, s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r, \boldsymbol{y}).$$

Examples include Metropolis–Hastings, Langevin, proximal, or other observation-correction kernels, depending on the likelihood model. By the factorization in (40), this correction can be interpreted as accounting for the residual likelihood factor

$$\ell(\boldsymbol{x}_0)^{\lambda_r - \nu}.$$

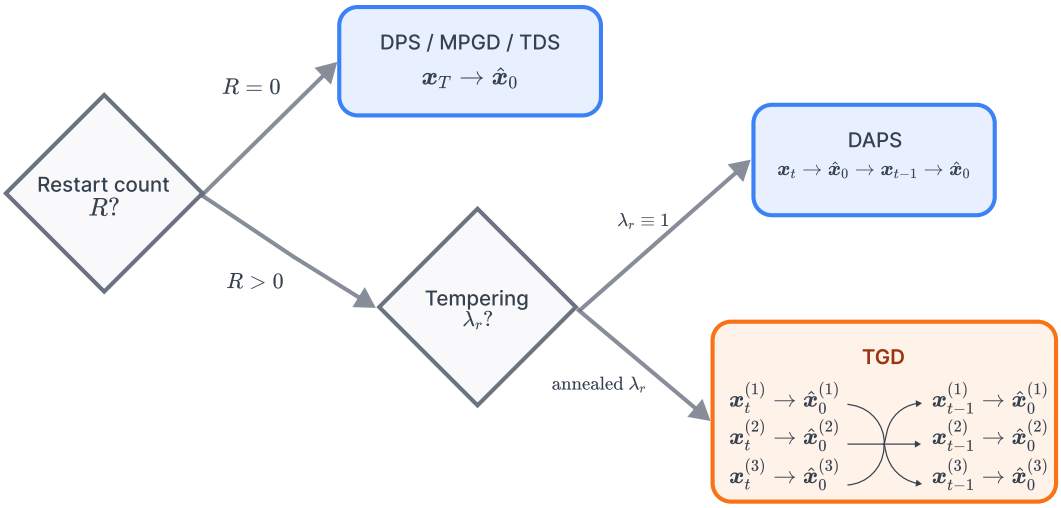


Figure 5: Settings under which existing training-free conditional samplers are recovered as special cases of TGD. Single-stage TGD recovers the chosen conditional reconstruction module, while fully conditioned multi-stage TGD with constant tempering recovers DAPS-style repeated reconstruction and correction.

More precisely, the full target is still $p_{\lambda_r,s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r, \boldsymbol{y})$; the residual factor describes how this target differs from the weaker law $p_{\nu,s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r, \boldsymbol{y})$. In the special case where exact samples from p_{ν,s_r} are used as an independence proposal, the Metropolis–Hastings acceptance ratio depends only on the residual likelihood factor. For generic local correction kernels, the transition must target the full stagewise density.

The endpoints of this construction recover the two reconstruction families used in the main text. If $\nu = 0$, then

$$p_{0,s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r, \boldsymbol{y}) = p_{s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r),$$

so the first step is unconditional reconstruction followed by clean-space observation correction, corresponding to a DAPS-style module. If $\nu = \lambda_r$, the residual factor is constant, and the clean-space correction step is unnecessary when the conditional reverse-dynamics reconstruction is exact. This recovers a fully conditional reverse-dynamics module. Intermediate values $0 < \nu < \lambda_r$ yield hybrid modules in which conditional reverse dynamics absorb part of the observation likelihood and the clean-space correction accounts for the remaining conditioning.

This hybrid construction is useful conceptually because it shows that conditional reverse dynamics and clean-space correction are compatible within the same TGD reconstruction interface. It also separates two sources of approximation: the accuracy of the weak conditional reconstruction and the mixing or accuracy of the clean-space correction kernel. In our preliminary experiments, however, hybrid choices with $0 < \nu < \lambda_r$ did not yield consistent improvements over the simpler endpoint modules used in the main experiments. We therefore report the endpoint variants in the main paper and include the hybrid construction here for completeness.

D Recovering Existing Training-Free Conditional Samplers

We describe how several existing training-free conditional diffusion samplers are recovered as special cases of TGD. In the notation of Sections 3.1–3.3, TGD is specified by an outer noise schedule $\{s_r\}_{r=0}^R$, a tempering schedule $\{\lambda_r\}_{r=0}^R$, a number of outer particles N, and a stagewise reconstruction module approximating

$$p_{\lambda_r,s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r, \boldsymbol{y}) \propto p_{s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r)\, p(\boldsymbol{y} \mid \boldsymbol{x}_0)^{\lambda_r}. \tag{41}$$

The corresponding clean-space annealed targets are

$$\pi_r(\boldsymbol{x}_0) \propto p(\boldsymbol{x}_0)\, p(\boldsymbol{y} \mid \boldsymbol{x}_0)^{\lambda_r}, \qquad 0 \le \lambda_R \le \cdots \le \lambda_0 = 1, \tag{42}$$

and the outer incremental weight update is

$$\widetilde{w}_{r-1}^{(i)} = w_r^{(i)} p(\boldsymbol{y} \mid \boldsymbol{x}_0^{(i)})^{\lambda_{r-1}-\lambda_r}, \qquad r = R, \dots, 1. \tag{43}$$

Here R denotes the number of outer transitions, while $r = 0$ denotes the terminal fully conditioned reconstruction stage.

Single-stage conditional samplers. When $R = 0$, TGD has no outer reweighting, resampling, or propagation stages. The tempering schedule consists only of the terminal level $\lambda_0 = 1$, and the algorithm initializes

$$\boldsymbol{z}_0 \sim p_{s_0}.$$

With the standard choice $s_0 = S$, this gives a single fully conditioned reconstruction from the highest-noise prior state:

$$\boldsymbol{x}_0 \sim p_{\lambda_0, s_0}(\boldsymbol{x}_0 \mid \boldsymbol{z}_0, \boldsymbol{y}) = p_{1,S}(\boldsymbol{x}_0 \mid \boldsymbol{z}_0, \boldsymbol{y}). \tag{44}$$

Therefore, TGD with $R = 0$ exactly recovers any chosen single-call training-free conditional sampler when that sampler is used as the reconstruction module. Choosing a DPS-style reconstruction module recovers DPS; choosing an MPGD-style reconstruction module recovers MPGD; choosing a TDS reconstruction module recovers TDS. In the TDS case, the particles and weights internal to TDS are part of the reconstruction module itself and are distinct from the outer TGD particle system.

DAPS-style multi-stage reconstruction. DAPS-style methods use repeated reconstruction and observation correction across a decreasing sequence of noise levels. This behavior is recovered by choosing a multi-stage noise schedule

$$S = s_R \geq s_{R-1} \geq \cdots \geq s_0 > 0, \qquad R \geq 1, \tag{45}$$

and setting the tempering schedule to be fully conditioned at every stage:

$$\lambda_r \equiv 1, \qquad r = 0, \dots, R. \tag{46}$$

Then, for every nonterminal stage $r = R, \dots, 1$,

$$\lambda_{r-1} - \lambda_r = 0,$$

so the incremental potential in (43) is constant. Consequently, the outer weighting step is vacuous. If resampling is omitted, and the reconstruction module at each stage is chosen to be the DAPS-style module consisting of unconditional reconstruction followed by clean-space observation correction, then TGD exactly recovers DAPS along the same noise schedule $\{s_r\}_{r=0}^{R}$.

For $N = 1$, this gives a single DAPS trajectory. For $N > 1$, omitting resampling gives independent multi-trajectory DAPS sampling. This corresponds to best-of-N baselines, which run several independent trajectories and select the final reconstruction using a data-consistency score. In contrast, TGD with a nonconstant tempering schedule introduces nontrivial likelihood-ratio weights and optional resampling, allowing outer particles to interact before the terminal reconstruction.

These reductions clarify the role of TGD. When the outer structure collapses to a single stage, TGD reduces to the chosen conditional reconstruction solver. When the tempering schedule is constant and fully conditioned, the likelihood-ratio weights are constant and the method reduces to repeated reconstruction and correction without outer particle interaction. The distinctive TGD regime is the annealed multi-stage setting: outer particles are reconstructed under progressively stronger conditioning, reweighted by likelihood-ratio increments, optionally resampled, and propagated across noise levels. This is the regime that allows computation to concentrate on trajectories that remain plausible under both the diffusion prior and the observation.

E Accelerated TGD pseudocode

Algorithm 2 uses the following terminal-selection convention. At $\rho = 1$, all N particles are propagated through every outer stage and reconstructed at the terminal level. The candidate with highest observation likelihood is then returned, equivalently the candidate with lowest measurement error under the Gaussian observation model. With the same reconstruction module, schedules, and resampling policy, this corresponds to full, unpruned TGD followed by terminal selection for a single reconstruction. This selected reconstruction is distinct from the weighted particle approximation returned by Algorithm 1.

Algorithm 2 Accelerated Tempered Guided Diffusion (A-TGD)

Require: observation $\boldsymbol{y}$, number of particles N, tempering schedule $\{\lambda_r\}_{r=0}^{R}$, noise schedule $\{s_r\}_{r=0}^{R}$, conditional reconstruction law $p_{\lambda_r,s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r, \boldsymbol{y})$, pruning factor $\rho \in [0,1]$

$K_\rho \leftarrow \min\{R, \max\{1, \lceil \rho R \rceil\}\}$
Sample $\boldsymbol{z}_R^{(i)} \sim p_{s_R}(\boldsymbol{z}_R)$ and set $w_R^{(i)} \leftarrow 1/N$, for $i = 1, \dots, N$
for $r = R, R-1, \dots, R-K_\rho+1$ **do**
  for $i = 1, \dots, N$ **do**
    Sample $\boldsymbol{x}_0^{(i)} \sim p_{\lambda_r,s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r^{(i)}, \boldsymbol{y})$
    $\tilde{w}_{r-1}^{(i)} \leftarrow w_r^{(i)} p(\boldsymbol{y} \mid \boldsymbol{x}_0^{(i)})^{\lambda_{r-1}-\lambda_r}$
  end for
  Normalize $\{\tilde{w}_{r-1}^{(i)}\}_{i=1}^{N}$ to obtain $\{w_{r-1}^{(i)}\}_{i=1}^{N}$
  Optionally resample according to $\{w_{r-1}^{(i)}\}_{i=1}^{N}$ and reset weights to $1/N$
  for $i = 1, \dots, N$ **do**
    Sample $\boldsymbol{z}_{r-1}^{(i)} \sim p_{s_{r-1}}(\boldsymbol{z}_{r-1} \mid \boldsymbol{x}_0^{(i)})$
  end for
end for
// Prune current particle population
for $i = 1, \dots, N$ **do**
  Sample $\bar{\boldsymbol{x}}_0^{(i)} \sim p_{\lambda_{R-K_\rho},s_{R-K_\rho}}(\boldsymbol{x}_0 \mid \boldsymbol{z}_{R-K_\rho}^{(i)}, \boldsymbol{y})$
end for
$i^\star \leftarrow \arg\max_i p(\boldsymbol{y} \mid \bar{\boldsymbol{x}}_0^{(i)})$
Set $\boldsymbol{z}_{R-K_\rho} \leftarrow \boldsymbol{z}_{R-K_\rho}^{(i^\star)}$
for $r = R-K_\rho, R-K_\rho-1, \dots, 1$ **do**
  Sample $\boldsymbol{x}_0 \sim p_{\lambda_r,s_r}(\boldsymbol{x}_0 \mid \boldsymbol{z}_r, \boldsymbol{y})$
  Sample $\boldsymbol{z}_{r-1} \sim p_{s_{r-1}}(\boldsymbol{z}_{r-1} \mid \boldsymbol{x}_0)$
end for
Sample $\boldsymbol{x}_0 \sim p_{\lambda_0,s_0}(\boldsymbol{x}_0 \mid \boldsymbol{z}_0, \boldsymbol{y})$
return $\boldsymbol{x}_0$

F Experimental and Implementation Details

F.1 Controlled inverse problems

We use a controlled two-dimensional inverse problem in which the prior, score, likelihood, and posterior reference sampler are all available analytically. This lets us evaluate the outer particle procedure without learned-score error.

Prior and analytic score. The clean prior is an equally weighted mixture of 5 Gaussians in $\mathbb{R}^2$. The component means are sampled uniformly from $[-0.9, 0.9]^2$, corresponding to a margin parameter 0.10, and each component has covariance $\tau^2 I$ with $\tau = 0.005$. We use an analytic diffusion model: under additive Gaussian noising at noise level s, the noised marginal remains a Gaussian mixture with component covariances $\tau^2 I + s^2 I$, and the score is computed exactly from this noised mixture distribution.

Observation model and posterior reference samples. For each of 10 test conditions, we sample a clean point $\boldsymbol{x}_0$ from the mixture prior and generate an elementwise absolute-value observation

$$\boldsymbol{y} = |\boldsymbol{x}_0| + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, 0.01^2 I).$$

The same observations are used for all methods. For each observation, we generate 10,000 reference posterior samples from the exact conditional distribution $p(\boldsymbol{x}_0 \mid \boldsymbol{y})$. This sampler exploits the diagonal Gaussian-mixture prior and the separable absolute-value likelihood: it first samples the mixture component conditional on $\boldsymbol{y}$, then samples each coordinate from the corresponding one-dimensional conditional distribution, which is a mixture over the two possible signs.

Samplers. All methods use the same analytic prior score and the same DPS-style conditional drift inside their reconstruction solver. The proposal likelihood used for DPS-style guidance has standard

deviation

$$\sigma_{\mathrm{prop}}(s) = 0.8s + 0.01,$$

while TGD particle weights use the true measurement likelihood with noise standard deviation 0.01.

TGD uses a 20-point EDM outer noise grid with highest noise $s_R = 80$, lowest positive noise $s_0 = 0.002$, and EDM curvature parameter 7. Its likelihood tempering schedule is uniform from $\lambda_R = 0$ to $\lambda_0 = 1$. Each outer stage uses a one-step Euler-discretized DPS-style ODE reconstruction proposal, followed by likelihood weighting, systematic resampling after nontrivial weighting steps, and EDM re-noising.

We compare against two baselines implemented in the same sampler framework. DPS uses a single outer stage and a 20-step Euler ODE DPS-style reconstruction trajectory, with no outer particle interaction. DPS-DAPS uses the same 20-point outer EDM noise grid as TGD but sets $\lambda_r = 1$ at every stage. Since the tempering exponent is constant, the incremental likelihood weights are constant, so no resampling is triggered. This baseline corresponds to repeated DPS/DAPS-style reconstruction without annealed SMC weighting or particle interaction.

Particle sweep and metrics. We sweep the logical particle count

$$N \in \{1, 2, 4, 8, 16, 32, 64, 128\}.$$

For each method and particle count, we run the corresponding N-particle sampler repeatedly and pool $10{,}000$ generated samples per observation before computing distributional metrics. For computational efficiency, the $N = 1$ pooled-sample setting is generated in larger independent batches, but it is evaluated as the logical single-particle setting.

We report the max-sliced Wasserstein distance, labeled as SWD in Figure 2, between generated samples and exact posterior reference samples. The metric is computed with the POT implementation [16, 17] per observation using 100 random projections, and then summarized over the 10 test conditions. Figure 2 reports means over the 10 test conditions, with bands corresponding to one standard error.

Visualization. For Figure 2(a), we visualize one representative test condition. Prior contours are estimated from $20{,}000$ prior samples, and posterior contours are estimated from exact posterior reference samples for the same observation. We overlay the final $N = 64$ TGD particles, the ground-truth clean point, and the sign-ambiguous candidate locations induced by the absolute-value observation.

Twenty-dimensional experiment. We extend the Gaussian-mixture inverse problem above to clean signals $x_0 \in \mathbb{R}^{20}$, retaining five diagonal Gaussian components and the separable absolute-value observation model. Exact-TGD uses exact draws from the stagewise reconstruction law (7), rather than the DPS approximation. The baselines use the analytic prior score.

We evaluate $N \in \{4, 8, 16, 32, 64, 128\}$ over ten test conditions. For each N, repeated independent N-particle runs are pooled to obtain 10,000 samples per condition; N remains the actual interacting particle count. We report mean marginal W_1, obtained by averaging one-dimensional Wasserstein distances over coordinates, and max-sliced W_1, computed using 1,000 fixed random projections shared across methods. Component TV measures error in the mixture-component probabilities. Distributional errors are summarized as means and standard errors across conditions.

The Monte Carlo reference error compares independent sets of 10,000 and 20,000 exact-posterior samples per condition. For Exact-TGD and TDS, normalized effective sample size, ESS/N, is measured immediately before resampling. We report its minimum stagewise mean.

Exact-TGD achieves distributional errors close to the Monte Carlo reference error. Although its minimum normalized ESS decreases with N, the corresponding absolute ESS increases from approximately 1.3 particles at $N = 4$ to 25.2 at $N = 128$. The posterior is numerically concentrated on one mixture component in this setting, so zero component TV does not establish within-component accuracy; the marginal and max-sliced distances remain informative.

Compute. The controlled experiments were run on CPU and completed in a few minutes.

Table 3: Posterior diagnostics for the twenty-dimensional inverse problem. Distributional errors are reported as mean ± standard error across 10 conditions. The reference floor compares 10,000 and 20,000 independent exact-posterior draws per condition. ESS is the minimum-stage mean pre-resampling ESS/N.

Method	N	Mean marginal $W_1 \downarrow$	Max-sliced $W_1 \downarrow$	Component TV↓	Min. ESS/N
Reference floor	–	$6.734\times10^{-5}\pm 1.21\times10^{-6}$	$1.546\times10^{-4}\pm 6.12\times10^{-6}$	0	–
Exact-TGD	4	$8.675\times10^{-5}\pm 3.06\times10^{-6}$	$2.323\times10^{-4}\pm 1.10\times10^{-5}$	0	.315
Exact-TGD	16	$7.225\times10^{-5}\pm 1.22\times10^{-6}$	$1.783\times10^{-4}\pm 8.96\times10^{-6}$	0	.201
Exact-TGD	128	$7.270\times10^{-5}\pm 2.46\times10^{-6}$	$1.742\times10^{-4}\pm 9.98\times10^{-6}$	0	.197
TDS	4	.1764± .0126	.6017± .0521	.2722± .0148	.456
TDS	16	.00617± .00120	.02141± .00459	.00832± .00173	.296
TDS	128	$7.718\times10^{-4}\pm 2.44\times10^{-5}$	$2.038\times10^{-3}\pm 1.08\times10^{-4}$	0	.145
DPS	128	.4014± .0016	1.288± .028	.656± .023	–
DPS-DAPS	128	.3831± .0098	1.237± .050	.612± .014	–

F.2 Noise and ODE discretization

We use the EDM noise discretization [24], following DAPS [55]. Our implementation uses two discretizations. The first is the outer TGD discretization, which defines the noise levels $s_R, \ldots, s_0$ at which particles are reconstructed, likelihood-reweighted, optionally resampled, and re-noised. For all image experiments, we use the same outer EDM discretization as DAPS, with maximum noise $s_R = 100$, minimum positive noise $s_0 = 0.1$, and curvature parameter 7.

The second is the inner ODE discretization used inside each reconstruction module. We denote the inner ODE variable by τ to distinguish it from the outer TGD levels s_r. All image experiments use four Euler ODE steps per reconstruction module, with minimum positive inner noise $\tau_{\min} = 0.01$ and curvature parameter 7.

F.3 Inverse problem setup

We use the same inverse-problem definitions and preprocessing as DAPS [55], which follows the DPS benchmark setup [8]. In all image experiments, the observation noise standard deviation is $\sigma = 0.05$.

For inpainting, the forward operator $\mathcal{A}$ applies the prescribed pixel mask to the clean image, and the observation is generated as

$$\boldsymbol{y} = \mathcal{A}(\boldsymbol{x}_0) + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}).$$

We use the same mask-generation protocol as DAPS and fix the mask realization across methods for fair comparison.

For phase retrieval, we use the normalized observation model from DAPS:

$$\boldsymbol{y} \sim \mathcal{N}\left(\left|\boldsymbol{F}\boldsymbol{P}\left(0.5\boldsymbol{x}_0 + 0.5\right)\right|, \sigma^2 \boldsymbol{I}\right),$$

where $\boldsymbol{F}$ is the discrete Fourier transform and $\boldsymbol{P}$ is the oversampling operator. Following DAPS, we use oversampling factor $k = 2$ and $n = 8$. The affine transformation $0.5\boldsymbol{x}_0 + 0.5$ maps images from $[-1, 1]$ to $[0, 1]$ before applying the phase-retrieval operator.

F.4 Pretrained diffusion models

We use the same pretrained diffusion models as DAPS [55]. For FFHQ, we use the 256×256 model used in DPS [8]. For ImageNet, we use the 256×256 model from [12]. All image inverse-problem generation and evaluation protocols follow DAPS [55], except that DAPS reports oracle best-of-N scores using the trajectory with the best LPIPS, PSNR, or SSIM against the ground truth, whereas we select the trajectory with the lowest measurement error. Since the ground-truth image is unavailable in practical inverse problems, oracle LPIPS/PSNR/SSIM selection is not possible at test time, while measurement-error selection depends only on the observation.

F.5 Existing assets and licenses

We use existing datasets, pretrained models, baseline implementations, and evaluation libraries only for research evaluation. FFHQ [23] is used under the Creative Commons BY-NC-SA 4.0

license. ImageNet [11] is used under the ImageNet terms of access for non-commercial research and educational use. We do not redistribute FFHQ, ImageNet, or pretrained diffusion checkpoints in the supplemental material; the reproduction instructions direct users to obtain these assets from their original sources and follow the corresponding terms of use.

Our implementation also depends on standard open-source scientific-computing and evaluation packages, including PyTorch, LPIPS [56], and POT [16, 17]. The supplemental code includes only materials that we are permitted to redistribute, and all third-party assets are credited through the corresponding citations and original access instructions.

F.6 Reconstruction modules

We instantiate TGD with DPS-, MPGD-, and DAPS-style reconstruction modules. Each module maps a noisy auxiliary state $\boldsymbol{x}_s$ at noise level s to an approximate clean reconstruction, which is then passed to the outer TGD likelihood-weighting, resampling, pruning, and re-noising steps. For the DPS- and MPGD-style modules, we use the conditional-score decomposition in (5) and approximate the intractable conditional term using the denoised estimate $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_s, s)$.

For DPS, we use the standard plug-in approximation

$$\nabla_{\boldsymbol{x}_s} \log p_s(\boldsymbol{y} \mid \boldsymbol{x}_s) \approx \nabla_{\boldsymbol{x}_s} \log \hat{p}_s(\boldsymbol{y} \mid \hat{\boldsymbol{x}}_0(\boldsymbol{x}_s, s)) \,.$$

For MPGD, we instead use a clean-space guidance term

$$\nabla_{\boldsymbol{x}_s} \log p_s(\boldsymbol{y} \mid \boldsymbol{x}_s) \approx \nabla_{\boldsymbol{x}_0} \log \hat{p}_s(\boldsymbol{y} \mid \hat{\boldsymbol{x}}_0(\boldsymbol{x}_s, s)) \,,$$

which is inserted into the MPGD reconstruction update. In both cases, the approximate proposal likelihood used inside the reconstruction module is

$$\hat{p}_s(\boldsymbol{y} \mid \boldsymbol{x}) = \mathcal{N}\big(\boldsymbol{y}; \mathcal{A}(\boldsymbol{x}), (\gamma s + \sigma)^2 \boldsymbol{I}\big) \,,$$

where $\mathcal{A}$ is the task forward operator, σ is the observation noise, and γ is a proposal-smoothing parameter that inflates the likelihood variance at high diffusion noise levels. This smoothed likelihood is used inside the reconstruction proposal. The outer TGD particle weights use the true measurement likelihood $p(\boldsymbol{y} \mid \boldsymbol{x}_0)$. For pruning, we select the particle with smallest measurement-space error,

$$i^\star = \arg\min_i \|\mathcal{A}(\hat{\boldsymbol{x}}_0^{(i)}) - \boldsymbol{y}\|_2^2,$$

which is equivalent to maximum likelihood selection under a Gaussian observation model with fixed variance. We use $\sigma = 0.05$ in all image experiments, with $\gamma = 0.7$ for inpainting and $\gamma = 0.4$ for phase retrieval.

For DAPS-style reconstruction, we follow the DAPS implementation [55]. In particular, DAPS uses 100 Langevin correction steps with a linearly decayed step size and the correction-noise setting reported in that work. This DAPS-style module is used for the phase retrieval A-TGD experiments and for the DAPS baselines.

F.7 Annealing schedules

TGD uses a sequence of likelihood-tempering exponents

$$\lambda_R, \lambda_{R-1}, \ldots, \lambda_0$$

to move from weakly conditioned targets to the final posterior. Stage $r = R$ is the earliest, highest-noise stage, while stage $r = 0$ is the final, lowest-noise stage. We require

$$0 \le \lambda_R \le \lambda_{R-1} \le \cdots \le \lambda_0 = 1,$$

so the final target corresponds to the full posterior.

Uniform schedule. The uniform schedule increases the likelihood exponent linearly across outer stages:

$$\lambda_r = \lambda_R + \frac{R - r}{R} (1 - \lambda_R) \,, \qquad r = 0, \ldots, R.$$

Thus the initial stage uses exponent λ_R, while the final stage uses $\lambda_0 = 1$. Setting $\lambda_R = 1$ gives the no-annealing variant, in which the full likelihood is applied at every stage.

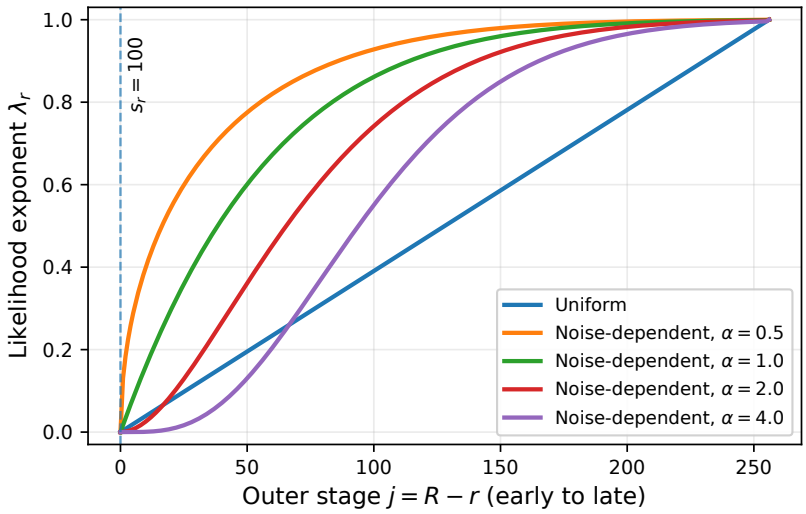


Figure 6: Likelihood-tempering schedules for the EDM discretization with 256 steps. The uniform schedule increases the exponent linearly across outer stages, while the noise-dependent schedules tie the exponent to the EDM noise level s_r.

Noise-dependent schedule. We also consider a schedule that ties the likelihood exponent directly to the diffusion noise level. Let s_r denote the noise level at stage r, ordered as

$$s_R \geq s_{R-1} \geq \cdots \geq s_0.$$

For a curvature parameter $\alpha > 0$, we define the normalized noise progress

$$u_r = \left(\frac{s_R - s_r}{s_R - 0} \right)^{\alpha},$$

and set the likelihood exponent to

$$\lambda_r = \lambda_R + (1 - \lambda_R) \left(\frac{s_R - s_r}{s_R - 0} \right)^{\alpha}.$$

This gives $u_R = 0$ and $u_0 = 1$, so the schedule starts from the initial likelihood exponent λ_R at the highest noise level and reaches the full likelihood exponent 1 at the lowest noise level. The parameter α controls the curvature of the transition: $\alpha = 1$ gives a linear transition in the noise level, while larger values keep λ_r closer to λ_R until lower noise levels. In our experiments, we use $s_R = 100$. Figure 6 visualizes the uniform schedule and several noise-dependent schedules for the EDM discretization with 256 steps.

F.8 Hyperparameter selection

All image-experiment hyperparameters were selected using a small tuning set of five images that was disjoint from both the diffusion-model training data and the 100-image validation set used for reporting results. After selecting the hyperparameters on this tuning set, we fixed them for all reported evaluations. The reported PSNR, SSIM, LPIPS, and runtime values were computed only on the held-out validation images and were not used for hyperparameter selection.

F.9 Image experiment configurations

Table 4 summarizes the A-TGD configurations used in the main image experiments. All main image experiments use $N = 4$ initial particles. For inpainting, we use an MPGD-style reconstruction module, uniform likelihood tempering, and systematic resampling after every likelihood-weighting step. For phase retrieval, we use a DAPS-style reconstruction module and no likelihood tempering, i.e. $\lambda_r = 1$ for all stages; therefore the incremental weights are constant and no resampling is performed.

Table 4: Main A-TGD configurations for image inverse problems. All settings use $N = 4$ particles. When resampling is enabled, we use systematic resampling.

Task	Dataset	Module	ρ	Annealing	Resampling
Inpainting	FFHQ	MPGD	0.5	Uniform	Always
Inpainting	ImageNet	MPGD	0.5	Uniform	Always
Phase retrieval	FFHQ	DAPS	0.5	None, $\lambda_r = 1$	Never
Phase retrieval	ImageNet	DAPS	0.8	None, $\lambda_r = 1$	Never

Table 5: Matched-runtime image sampling configurations.

Task	Dataset	Method	NFE scheme	Time/img (s)
Inpainting	FFHQ	DPS	0×596	6.34
		DAPS ($N = 1$)	299×4	6.44
		DAPS ($N = 4$)	77×4	6.50
		A-TGD	128×4	6.47
	ImageNet	DPS	0×596	23.63
		DAPS ($N = 1$)	320×4	23.52
		DAPS ($N = 4$)	80×4	23.84
		A-TGD	128×4	23.64
Phase retrieval	FFHQ	DPS	0×404	4.34
		DAPS ($N = 1$)	159×4	4.30
		DAPS ($N = 4$)	41×4	4.35
		A-TGD	64×4	4.35
	ImageNet	DPS	0×7860	346
		DAPS ($N = 1$)	3855×4	357
		DAPS ($N = 4$)	1070×4	352
		A-TGD	1250×4	345

The pruning factor is $\rho = 0.5$ for all image experiments except ImageNet phase retrieval, where we use $\rho = 0.8$.

When resampling is enabled, we use systematic resampling and reset the particle weights to $1/N$. Our implementation also supports effective-sample-size- (ESS) threshold resampling, but we do not use it in the main experiments. In preliminary experiments, ESS-threshold resampling produced similar behavior to the simpler always-resample or never-resample policies, while introducing an additional task-dependent hyperparameter. We therefore use the simpler policies in Table 4 to keep the matched-runtime comparison transparent.

F.10 Matched-runtime image configurations

Table 5 reports the sampling configurations used for the matched-runtime image experiments. For ODE-based methods, the NFE scheme is written as the number of reconstruction calls times the four ODE steps used per reconstruction. For DPS, which does not use the restart-based reconstruction interface, we write $0 \times K$, where K is the total number of function evaluations. For DAPS ($N = 4$), the NFE scheme is reported per trajectory; the reported wall-clock time includes all four trajectories. Wall-clock time per image is measured by dividing the total sampling time for the 100-image evaluation set by 100.

F.11 Speed-quality tradeoff configuration

Figure 3 evaluates the speed-quality tradeoff on FFHQ phase retrieval. For each method, we sweep the sampling budget and record reconstruction quality on 100 validation images. Wall-clock time per image is measured by dividing the total sampling time for the 100-image evaluation set by 100, including all independent trajectories or particle computations.

Table 6 reports the configurations used to generate the curves. For ODE-based methods, the NFE scheme is reported as the number of reconstruction calls times the four ODE steps used per recon-

Table 6: FFHQ phase retrieval speed-quality sweep.

Method	NFE scheme	Time/img (s) ↓	LPIPS ↓	PSNR ↑
DPS	0×68	0.73	$0.561_{\pm 0.063}$	$12.67_{\pm 2.42}$
	0×104	1.12	$0.562_{\pm 0.063}$	$12.67_{\pm 2.42}$
	0×204	2.18	$0.563_{\pm 0.063}$	$12.66_{\pm 2.42}$
	0×404	4.31	$0.564_{\pm 0.063}$	$12.65_{\pm 2.42}$
	0×800	8.53	$0.564_{\pm 0.063}$	$12.65_{\pm 2.42}$
	0×1592	16.97	$0.564_{\pm 0.063}$	$12.65_{\pm 2.42}$
DAPS ($N = 1$)	26×4	0.70	$0.427_{\pm 0.191}$	$19.18_{\pm 7.40}$
	42×4	1.13	$0.354_{\pm 0.223}$	$21.95_{\pm 8.54}$
	81×4	2.18	$0.245_{\pm 0.173}$	$25.67_{\pm 8.19}$
	159×4	4.29	$0.225_{\pm 0.185}$	$26.98_{\pm 8.35}$
	315×4	8.51	$0.208_{\pm 0.175}$	$28.10_{\pm 8.08}$
	628×4	16.94	$0.225_{\pm 0.191}$	$27.52_{\pm 8.68}$
DAPS ($N = 4$)	7×4	0.74	$0.561_{\pm 0.099}$	$16.83_{\pm 4.08}$
	11×4	1.16	$0.478_{\pm 0.128}$	$18.89_{\pm 4.73}$
	21×4	2.21	$0.359_{\pm 0.160}$	$22.42_{\pm 6.10}$
	41×4	4.31	$0.207_{\pm 0.111}$	$27.81_{\pm 5.12}$
	81×4	8.51	$0.169_{\pm 0.116}$	$29.56_{\pm 5.31}$
	162×4	17.03	$0.135_{\pm 0.033}$	$31.40_{\pm 2.38}$
A-TGD	9×4	0.71	$0.508_{\pm 0.114}$	$18.13_{\pm 4.53}$
	16×4	1.14	$0.388_{\pm 0.143}$	$21.59_{\pm 5.38}$
	32×4	2.19	$0.259_{\pm 0.148}$	$25.62_{\pm 6.11}$
	64×4	4.30	$0.172_{\pm 0.106}$	$29.43_{\pm 4.07}$
	128×4	8.53	$0.153_{\pm 0.089}$	$30.24_{\pm 4.49}$
	256×4	16.97	$0.152_{\pm 0.091}$	$30.63_{\pm 4.60}$

struction. For DPS, which does not use the restart-based reconstruction interface, we write $0 \times K$ to indicate no reconstruction restarts and K total function evaluations.

For Figure 3b, we compare A-TGD and DAPS $(N = 4)$ at fixed LPIPS thresholds

$$\ell \in \{0.45, 0.40, 0.35, 0.30, 0.25, 0.20\}.$$

For each threshold, we estimate the wall-clock time required by each method to reach that LPIPS value by linearly interpolating its LPIPS–runtime curve from Table 6. The speedup is

$$\text{speedup}(\ell) = \frac{T_{\text{DAPS}(N=4)}(\ell)}{T_{\text{A-TGD}}(\ell)},$$

where $T_m(\ell)$ is the interpolated time per image required by method m to reach LPIPS threshold ℓ. Values greater than one indicate that A-TGD reaches the same LPIPS threshold faster than independent best-of-four DAPS sampling.

G Ablations

Unless otherwise stated, all ablations are run on FFHQ using 100 validation images and report mean $\pm$ standard deviation for PSNR, SSIM, LPIPS, and wall-clock time per image in seconds. We use $N = 4$, $\rho = 0.5$, 64 restarts and 4 ODE sampling steps with the MPGD as the default reconstruction module. For inpainting, the default SMC configuration uses the uniform annealing with resampling; for phase retrieval, it uses no annealing. These defaults are used only for the ablations in this section; the main experiments use the task-specific configurations described in Appendix F.

Number of particles. Table 7 varies the number of initial A-TGD particles N. For FFHQ inpainting, increasing N has only a small effect on reconstruction quality: performance saturates around $N = 4$, and larger particle counts mainly increase wall-clock time. For FFHQ phase retrieval, the effect of N

Table 7: Particle-count ablation for A-TGD on FFHQ.

Task	Particles	Time/img (s) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Inpainting	$N = 1$	1.29	$24.61_{\pm 2.70}$	$0.846_{\pm 0.026}$	$0.126_{\pm 0.028}$
	$N = 2$	1.92	$24.73_{\pm 2.34}$	$0.848_{\pm 0.024}$	$0.125_{\pm 0.028}$
	$N = 4$	3.26	$\mathbf{25.01}_{\pm 2.34}$	$\mathbf{0.849}_{\pm 0.025}$	$0.124_{\pm 0.028}$
	$N = 8$	5.85	$24.88_{\pm 2.28}$	$0.848_{\pm 0.025}$	$0.124_{\pm 0.028}$
	$N = 16$	11.14	$24.92_{\pm 2.45}$	$0.849_{\pm 0.025}$	$\mathbf{0.124}_{\pm 0.028}$
Phase Retrieval	$N = 1$	1.30	$22.26_{\pm 7.61}$	$0.678_{\pm 0.213}$	$0.303_{\pm 0.194}$
	$N = 2$	1.94	$24.76_{\pm 7.02}$	$0.734_{\pm 0.192}$	$0.248_{\pm 0.169}$
	$N = 4$	3.27	$26.38_{\pm 5.63}$	$0.778_{\pm 0.145}$	$0.206_{\pm 0.129}$
	$N = 8$	5.91	$27.00_{\pm 5.52}$	$0.793_{\pm 0.151}$	$0.193_{\pm 0.125}$
	$N = 16$	11.27	$\mathbf{27.24}_{\pm 4.85}$	$\mathbf{0.798}_{\pm 0.131}$	$\mathbf{0.189}_{\pm 0.110}$

Table 8: Pruning-fraction ablation for A-TGD on FFHQ.

Task	Pruning factor ρ	Time/img (s) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Inpainting	0.10	1.76	$24.88_{\pm 2.32}$	$0.848_{\pm 0.025}$	$\mathbf{0.123}_{\pm 0.028}$
	0.25	2.30	$24.65_{\pm 2.57}$	$0.847_{\pm 0.025}$	$0.125_{\pm 0.028}$
	0.50	3.26	$\mathbf{25.01}_{\pm 2.34}$	$\mathbf{0.849}_{\pm 0.025}$	$0.124_{\pm 0.028}$
	0.75	4.22	$24.64_{\pm 2.44}$	$0.845_{\pm 0.025}$	$0.126_{\pm 0.027}$
	1.00	5.12	$24.93_{\pm 2.41}$	$0.849_{\pm 0.024}$	$0.124_{\pm 0.029}$
Phase Retrieval	0.10	1.80	$23.25_{\pm 7.88}$	$0.688_{\pm 0.225}$	$0.292_{\pm 0.203}$
	0.25	2.33	$23.47_{\pm 7.82}$	$0.697_{\pm 0.219}$	$0.282_{\pm 0.194}$
	0.50	3.28	$26.38_{\pm 5.63}$	$0.778_{\pm 0.145}$	$0.206_{\pm 0.129}$
	0.75	4.23	$27.01_{\pm 5.13}$	$0.795_{\pm 0.145}$	$0.193_{\pm 0.124}$
	1.00	5.13	$\mathbf{27.19}_{\pm 5.13}$	$\mathbf{0.802}_{\pm 0.131}$	$\mathbf{0.188}_{\pm 0.111}$

is much stronger. Increasing N from 1 to 4 substantially improves PSNR, SSIM, and LPIPS, and larger values continue to improve quality, although with diminishing returns and increased runtime.

These results indicate that the value of particle exploration depends on the ambiguity of the inverse problem. Inpainting is comparatively stable, so a small particle population is sufficient. Phase retrieval exhibits higher trajectory variability, and therefore benefits more from maintaining multiple early particles before pruning.

Pruning fraction. Table 8 varies the pruning factor ρ, which controls how long A-TGD maintains multiple particles before pruning to a single trajectory. For inpainting, quality is relatively insensitive to ρ, and aggressive early pruning already gives strong performance. For phase retrieval, the pruning factor has a much larger effect: later pruning substantially improves PSNR, SSIM, and LPIPS, at the cost of increased runtime.

Annealing and resampling. Table 9 studies the effect of annealing and resampling. For this ablation, we set $\rho = 1.0$ to isolate the effect of annealing and resampling at matched compute across variants. "No annealing" sets $\lambda_r = 1$ for all stages, applying the full likelihood at every stage, while "no resampling" uses the annealed schedule but disables resampling. The results show that the preferred SMC configuration differs across tasks. For inpainting, annealing with resampling gives the best reconstruction quality across PSNR, SSIM, and LPIPS, although the margin over the ablated variants is modest. For phase retrieval, however, the no-annealing variant performs best across all metrics. We therefore use annealing with resampling as the default for FFHQ inpainting and no annealing as the default for FFHQ phase retrieval.

Reconstruction module. Table 10 compares different reconstruction modules inside the same A-TGD outer procedure. We use no annealing in this ablation. We use MPGD as the default reconstruction module because it provides the best wall-clock efficiency while remaining competitive in reconstruction quality. The best-quality module, however, depends on the task. For inpainting, DPS gives the strongest PSNR, SSIM, and LPIPS, but is substantially slower. For phase retrieval, DAPS gives the best reconstruction quality, while MPGD is faster and remains competitive.

Table 9: Ablation of annealing and resampling.

Task	Variant	Time/img (s) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Inpainting	No annealing	5.05	$24.36_{\pm 2.64}$	$0.843_{\pm 0.027}$	$0.130_{\pm 0.031}$
	No resampling	5.06	$24.42_{\pm 2.59}$	$0.845_{\pm 0.027}$	$0.127_{\pm 0.031}$
	Annealing + resampling	5.05	$\mathbf{24.93}_{\pm 2.41}$	$\mathbf{0.849}_{\pm 0.024}$	$\mathbf{0.124}_{\pm 0.029}$
Phase Retrieval	No annealing	5.05	$\mathbf{27.19}_{\pm 5.13}$	$\mathbf{0.802}_{\pm 0.131}$	$\mathbf{0.188}_{\pm 0.111}$
	No resampling	5.06	$26.98_{\pm 5.23}$	$0.796_{\pm 0.135}$	$0.191_{\pm 0.113}$
	Annealing + resampling	5.07	$24.80_{\pm 7.79}$	$0.727_{\pm 0.217}$	$0.254_{\pm 0.190}$

Table 10: Reconstruction-module ablation inside the A-TGD outer loop.

Task	Reconstruction module	Time/img (s) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Inpainting	DPS	6.75	$\mathbf{25.66}_{\pm 2.42}$	$\mathbf{0.873}_{\pm 0.023}$	$\mathbf{0.112}_{\pm 0.024}$
	MPGD	**3.23**	$24.86_{\pm 2.35}$	$0.847_{\pm 0.025}$	$0.126_{\pm 0.028}$
	DAPS	3.41	$24.60_{\pm 2.00}$	$0.791_{\pm 0.027}$	$0.156_{\pm 0.034}$
Phase Retrieval	DPS	6.77	$26.38_{\pm 5.87}$	$0.782_{\pm 0.151}$	$0.217_{\pm 0.129}$
	MPGD	**3.27**	$26.38_{\pm 5.63}$	$0.778_{\pm 0.145}$	$0.206_{\pm 0.129}$
	DAPS	4.27	$\mathbf{29.43}_{\pm 4.07}$	$\mathbf{0.811}_{\pm 0.104}$	$\mathbf{0.172}_{\pm 0.106}$

The ablations show that A-TGD is most useful when early trajectories differ substantially in quality. Inpainting is relatively stable and benefits little from larger particle populations or later pruning, whereas phase retrieval benefits strongly from both. The reconstruction-module ablation confirms that the outer A-TGD procedure is modular: MPGD provides the fastest default, while DPS or DAPS can improve quality for specific tasks at higher cost. The annealing ablation shows that tempering is not uniformly beneficial: it helps modestly for inpainting but hurts phase retrieval in this setting.

Annealing schedule without resampling. Table 11 compares different annealing schedules for A-TGD without resampling. For inpainting, the choice of annealing schedule has only a small effect on reconstruction quality: uniform annealing gives the best SSIM and LPIPS, while the noise-dependent schedule with $\alpha = 0.5$ gives the best PSNR. In contrast, for phase retrieval, annealing does not improve performance. The no-annealing variant achieves the best PSNR, SSIM, and LPIPS, while stronger noise-dependent annealing progressively degrades reconstruction quality.

Annealing schedule with always resampling. Table 12 reports the same annealing-schedule ablation when resampling is performed at every annealing step. Compared to the no-resampling setting in Table 11, always resampling gives small improvements for inpainting. For phase retrieval, always resampling substantially degrades reconstruction quality across all annealing schedules: even the best resampling variant remains worse than the corresponding no-resampling result, and uniform annealing is particularly unstable. These results suggest that frequent resampling can remove useful particle diversity in harder inverse problems.

Table 11: Annealing-schedule ablation for A-TGD on FFHQ without resampling.

Task	Annealing schedule	Time/img (s) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Inpainting	No annealing	3.29	$24.86_{\pm 2.35}$	$0.847_{\pm 0.025}$	$0.126_{\pm 0.028}$
	Uniform annealing	3.27	$24.85_{\pm 2.34}$	$\mathbf{0.848}_{\pm 0.025}$	$\mathbf{0.124}_{\pm 0.028}$
	Noise-dependent, $\alpha = 0.5$	3.27	$\mathbf{24.87}_{\pm 2.27}$	$0.847_{\pm 0.024}$	$0.126_{\pm 0.028}$
	Noise-dependent, $\alpha = 1$	3.27	$24.85_{\pm 2.31}$	$0.847_{\pm 0.024}$	$0.126_{\pm 0.028}$
	Noise-dependent, $\alpha = 2$	3.27	$24.86_{\pm 2.25}$	$0.847_{\pm 0.023}$	$0.126_{\pm 0.027}$
	Noise-dependent, $\alpha = 4$	3.27	$24.82_{\pm 2.51}$	$0.847_{\pm 0.025}$	$0.126_{\pm 0.029}$
Phase Retrieval	No annealing	3.25	$\mathbf{26.38}_{\pm 5.63}$	$\mathbf{0.778}_{\pm 0.145}$	$\mathbf{0.206}_{\pm 0.129}$
	Uniform annealing	3.24	$22.92_{\pm 6.45}$	$0.689_{\pm 0.189}$	$0.284_{\pm 0.163}$
	Noise-dependent, $\alpha = 0.5$	3.24	$26.12_{\pm 5.80}$	$0.770_{\pm 0.153}$	$0.212_{\pm 0.133}$
	Noise-dependent, $\alpha = 1$	3.23	$26.01_{\pm 5.77}$	$0.770_{\pm 0.150}$	$0.211_{\pm 0.132}$
	Noise-dependent, $\alpha = 2$	3.24	$25.33_{\pm 6.47}$	$0.750_{\pm 0.173}$	$0.230_{\pm 0.155}$
	Noise-dependent, $\alpha = 4$	3.23	$25.03_{\pm 6.57}$	$0.747_{\pm 0.179}$	$0.236_{\pm 0.161}$

Table 12: Annealing-schedule ablation for A-TGD on FFHQ with always resampling.

Task	Annealing schedule	Time/img (s) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Inpainting	Uniform annealing	3.19	$25.01_{\pm 2.34}$	$\mathbf{0.849}_{\pm 0.025}$	$\mathbf{0.124}_{\pm 0.028}$
	Noise-dependent, $\alpha = 0.5$	3.19	$24.94_{\pm 2.21}$	$0.847_{\pm 0.024}$	$0.126_{\pm 0.028}$
	Noise-dependent, $\alpha = 1$	3.19	$24.90_{\pm 2.37}$	$0.848_{\pm 0.024}$	$0.125_{\pm 0.028}$
	Noise-dependent, $\alpha = 2$	3.19	$24.89_{\pm 2.38}$	$0.847_{\pm 0.024}$	$0.126_{\pm 0.029}$
	Noise-dependent, $\alpha = 4$	3.18	$\mathbf{25.05}_{\pm 2.26}$	$0.848_{\pm 0.024}$	$0.125_{\pm 0.028}$
Phase Retrieval	Uniform annealing	3.25	$21.66_{\pm 7.27}$	$0.654_{\pm 0.210}$	$0.318_{\pm 0.193}$
	Noise-dependent, $\alpha = 0.5$	3.25	$\mathbf{24.77}_{\pm 7.40}$	$\mathbf{0.735}_{\pm 0.198}$	$\mathbf{0.248}_{\pm 0.175}$
	Noise-dependent, $\alpha = 1$	3.24	$23.92_{\pm 7.79}$	$0.712_{\pm 0.208}$	$0.269_{\pm 0.186}$
	Noise-dependent, $\alpha = 2$	3.24	$24.18_{\pm 7.52}$	$0.719_{\pm 0.204}$	$0.267_{\pm 0.187}$
	Noise-dependent, $\alpha = 4$	3.24	$22.25_{\pm 7.86}$	$0.668_{\pm 0.220}$	$0.309_{\pm 0.206}$

Paired significance analysis. Table 13 reports paired comparisons of A-TGD with DAPS ($N = 4$) using measurements on the same 100 validation images for each task and dataset. We report the mean paired advantage Δ, defined as A-TGD minus DAPS for PSNR and SSIM, and DAPS minus A-TGD for LPIPS, so positive values always favour A-TGD. We obtain 95% confidence intervals using 100,000 paired bootstrap replicates and compute two-sided paired sign-flip permutation tests. The resulting p-values are Holm-corrected across the three metrics separately for each task and dataset.

The paired analysis finds statistically significant advantages for A-TGD across all three metrics on both FFHQ tasks, and for LPIPS and SSIM on ImageNet inpainting. The PSNR difference on ImageNet inpainting is not statistically significant ($p_{\mathrm{Holm}} = 0.261$), and none of the three differences on ImageNet phase retrieval is statistically significant ($p_{\mathrm{Holm}} = 1.000$).

H Broader Impacts

TGD and A-TGD are intended as methodological tools for training-free conditional diffusion and inverse-problem reconstruction. Potential positive impacts include more efficient use of pretrained diffusion priors for scientific imaging, restoration, and other inverse problems where paired task-specific training data are unavailable or costly. A-TGD may also reduce computational cost relative to independent multi-trajectory baselines by using multiple particles early and pruning to a single trajectory, improving speed-quality tradeoffs under fixed wall-clock budgets.

The same capabilities also have potential negative impacts. Improvements in image reconstruction and inpainting could be misused for image manipulation, misleading visual content generation, or reconstruction of sensitive visual information from incomplete measurements. These risks are partially mitigated by the scope of this work: we do not release a new pretrained generative model or dataset, and our experiments are limited to standard benchmark datasets and inverse-problem settings. Nevertheless, downstream deployments should consider data provenance, consent, privacy, and misuse monitoring, especially in applications involving people or sensitive imagery.

Table 13: Paired comparisons of A-TGD with DAPS ($N = 4$), using the evaluation results reported during rebuttal on 100 images per setting. Positive Δ favours A-TGD. Brackets contain 95% paired-bootstrap confidence intervals. The permutation-test p-values are Holm-corrected across the three metrics within each setting. "A-TGD wins" reports the percentage of images on which A-TGD achieves the better metric value.

Metric	A-TGD	DAPS-4	Δ [95% CI]	p_{Holm}	A-TGD wins
FFHQ inpainting					
PSNR ↑	25.1436	24.5204	$0.6232\ [0.3783, 0.8691]$	3.0×10^{-5}	63%
LPIPS ↓	0.1224	0.1568	$0.0344\ [0.0317, 0.0371]$	3.0×10^{-5}	100%
SSIM ↑	0.8525	0.7899	$0.0626\ [0.0589, 0.0663]$	3.0×10^{-5}	100%
ImageNet inpainting					
PSNR ↑	20.8902	20.7031	$0.1872\ [-0.1328, 0.5100]$	0.261	53%
LPIPS ↓	0.2041	0.2273	$0.0232\ [0.0172, 0.0292]$	3.0×10^{-5}	73%
SSIM ↑	0.7809	0.7402	$0.0407\ [0.0354, 0.0461]$	3.0×10^{-5}	96%
FFHQ phase retrieval					
PSNR ↑	29.4316	27.8105	$1.6211\ [0.9524, 2.3649]$	3.0×10^{-5}	83%
LPIPS ↓	0.1715	0.2068	$0.0352\ [0.0181, 0.0533]$	3.0×10^{-5}	92%
SSIM ↑	0.8111	0.7654	$0.0457\ [0.0277, 0.0655]$	3.0×10^{-5}	96%
ImageNet phase retrieval					
PSNR ↑	22.179	22.674	$-0.495\ [-2.286, 1.302]$	1.000	42%
LPIPS ↓	0.3256	0.3170	$-0.0086\ [-0.0553, 0.0382]$	1.000	44%
SSIM ↑	0.6069	0.6081	$-0.0012\ [-0.0543, 0.0527]$	1.000	46%

I Further Image Examples

We provide additional qualitative examples for each image inverse-problem setting. For each task and dataset, we visualize examples with the largest absolute LPIPS difference between A-TGD and DAPS ($N = 4$). These examples are intended to highlight cases where the methods differ most in perceptual reconstruction quality and are taken from the same evaluation runs used for the matched-runtime results in Table 1.

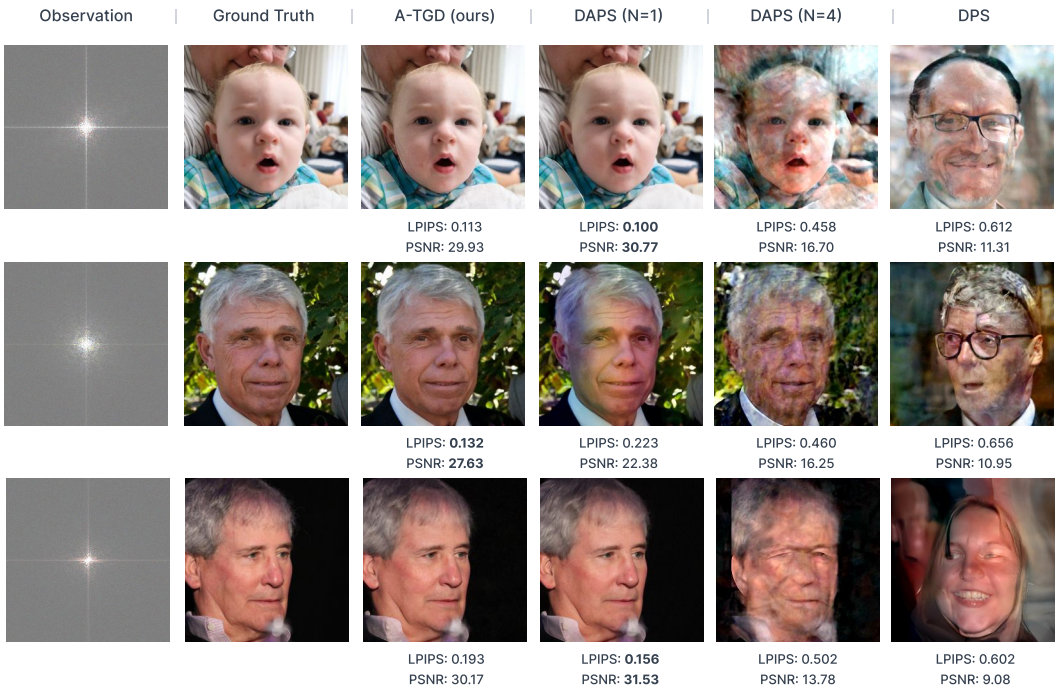


Figure 7: FFHQ phase retrieval examples.

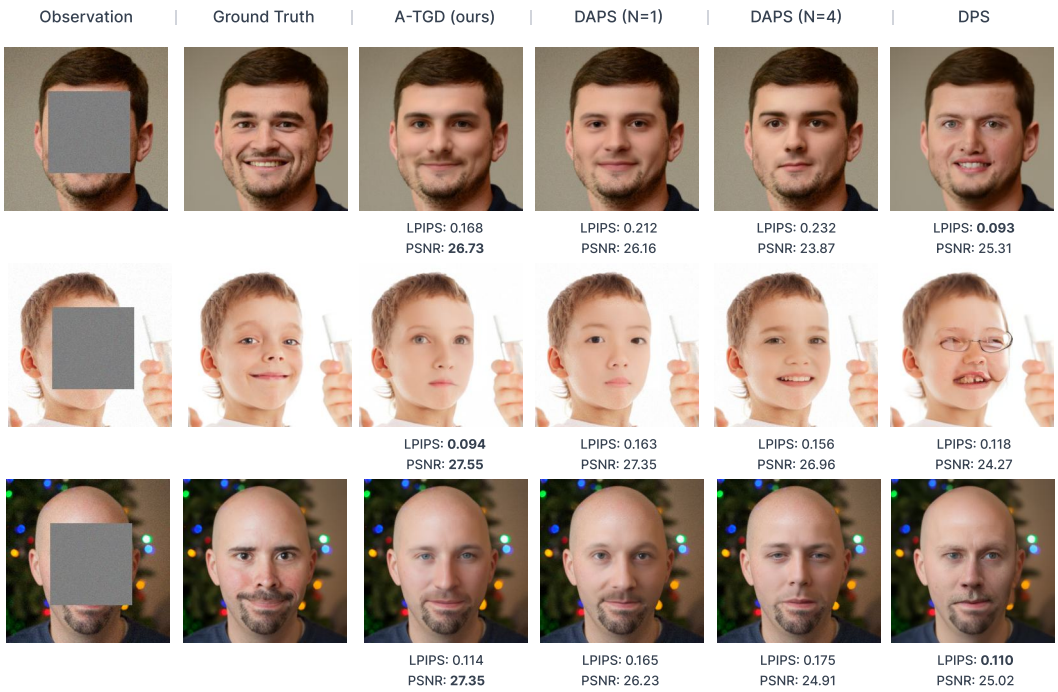


Figure 8: FFHQ inpainting examples.

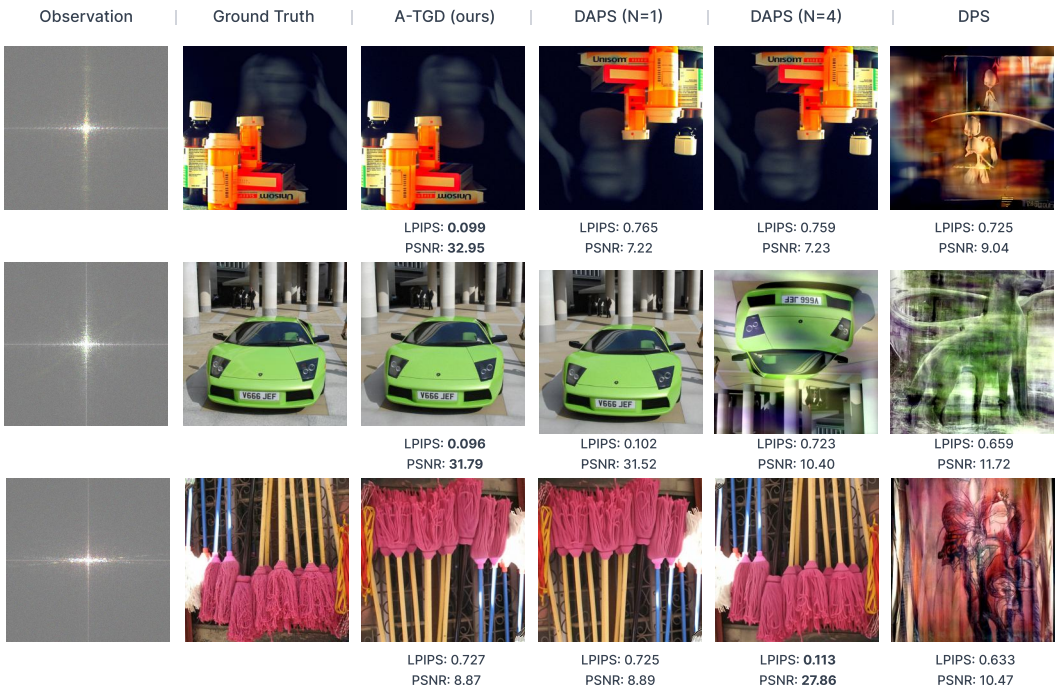


Figure 9: ImageNet phase retrieval examples.



Figure 10: ImageNet inpainting examples.

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 Section 1 state the main claims and scope, including TGD as a clean-space SMC framework, idealized consistency for full TGD, and A-TGD as a pruned reconstruction variant. These claims are supported by Sections 3–5 and qualified in Section 6.

   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: Section 6 discusses the main limitations, including the gap between ideal full-particle TGD and approximate/pruned image experiments, as well as settings where resampling or pruning may increase variance.

   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: The main theoretical result is stated in Section 3.4, with its assumptions and complete proof given in Appendix B; Appendix A also derives the incremental SMC weights used by the method.

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: Sections 5 and Appendix F describe the experimental setup, datasets, metrics, baselines, particle counts, sampling budgets, likelihood schedules, and wall-clock measurement procedure needed to reproduce the main results.

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 code and instructions in the supplemental material, including data preparation, environment setup, and scripts for reproducing the main 2D and image inverse-problem experiments.

   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: Section 5 and Appendix F specify the datasets, evaluation set size, metrics, baselines, particle counts, sampling budgets, likelihood schedules, pruning settings, and wall-clock measurement procedure. No model training or optimizer is used, since all methods use pretrained diffusion priors in a training-free setting.

   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: Figure 2 and Appendix F.1 report means with standard-error bands over 10 test conditions for the controlled 2D experiment. Section 5, Table 1, and Appendix F report image metrics as mean $\pm$ standard deviation over 100 validation images. Table 13 additionally reports 95% paired-bootstrap confidence intervals and two-sided paired sign-flip permutation tests, with Holm correction across the three metrics within each task and dataset.

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 and Appendix F state that image experiments use a single NVIDIA H200 GPU with 141GB memory, and Tables 5 and 6 report NFE schemes and wall-clock time per image. Appendix F.1 specifies the controlled experiments setup, which is run on CPU and completes in a few minutes.

   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: The authors have reviewed the NeurIPS Code of Ethics, and the research uses standard benchmark datasets and pretrained models in accordance with their terms of use, as described in Sections 5 and Appendix F.4.

   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: Appendix H discusses potential positive impacts, including more efficient inverse-problem reconstruction with pretrained diffusion priors, and potential negative impacts, including misuse for image manipulation or reconstruction of sensitive visual information.

    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: The paper does not release a new pretrained model, image generator, scraped dataset, or other high-risk asset; it releases only code and instructions, and uses existing benchmark datasets and pretrained models as described in Appendix F.4. Appendix H discusses the potential misuse risks and scope of the release.

    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: Sections 5 and Appendix F.4 cite the datasets, pretrained models, baseline methods, and evaluation libraries used in the experiments. Appendix F.5 states the FFHQ and ImageNet 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: The paper releases project code through a GitHub repository, with documentation including setup instructions, reproduction commands, and a license.

    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: The paper does not involve crowdsourcing, human annotators, user studies, or newly collected human-subject data.

    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: The paper does not involve crowdsourcing, human annotators, user studies, or newly collected human-subject data.

    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: [N/A].

    Justification: LLMs are not used as an important, original, or non-standard component of the method, experiments, data generation, evaluation, or analysis. Any LLM use was limited to writing, editing, formatting, or drafting and did not affect the core methodology, scientific rigor, or originality of the research.

    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.