BACKWARD-CONSISTENT DIFFUSION SAMPLING FOR SPARSELY OBSERVED PDE INVERSE PROBLEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recovering Partial Differential Equation (PDE) coefficient fields from extremely sparse observations is a severely ill-posed inverse problem for which generative machine learning methods (e.g., diffusion models) have become a leading way to encode the prior. Recent state-of-the-art diffusion solvers lift these priors to function spaces, finding a physics-consistent reconstruction in the output space of the diffusion denoiser. We prove that, in a discontinuous PDE setting, output space methods can result in failure to appropriately minimize the unobserved error with the correct coefficient field. Consequently, we propose **Fun**ction space **B**ackward-**C**onsistent **S**ampling (FunBCS), an *input space* optimization approach for solving PDE problems which aims to find the best input such that the denoiser reconstruction is physics-consistent. We then prove that FunBCS appropriately minimizes the unobserved error, unlike output space optimization methods. Per our theoretical analysis, we also provide insights on how to dynamically allocate the number of input space optimization steps used throughout the sampling process. Our evaluations, across four PDE inverse problems (including the discontinuous Darcy flow), demonstrate that FunBCS reduces the reconstruction error by 27–64% while running 1.4–2.1× faster when compared to the current state-of-the-art. Code is available at `https://anonymous.4open.science/r/FunBCS-7C24/`.

1 INTRODUCTION

Inverse problems governed by partial differential equations (PDEs) seek to estimate an unknown physical coefficient field $\boldsymbol{a}$ from sparse and sometimes noisy observations $\boldsymbol{u}_{\text{obs}}$. These problems are ubiquitous in observational science, where an incomplete observation must be described by a physical model, and arise in many applications including geophysics (Virieux & Operto, 2009; Bui-Thanh et al., 2013), astrophysics (Peng et al., 2024; Christensen & Meyer, 2022), biomedical imaging (Arridge, 1999; Poudel et al., 2019), and materials science (Bonnet & Constantinescu, 2005; Avril et al., 2008). Recovering the physical coefficient field $\boldsymbol{a}$ is *ill-posed*, since the observations $\boldsymbol{u}_{\text{obs}}$ often correspond to several possible coefficient fields $\boldsymbol{a}$. Thus, one aims to sample from the posterior $\pi(\boldsymbol{a}|\boldsymbol{u}_{\text{obs}})$ (Stuart, 2010), e.g., classical methods use Markov Chain Monte Carlo (MCMC), which can exhibit slow convergence for multimodal posteriors (Guan & Krone, 2007) or in high dimensions (Roberts et al., 1997). Additionally, practical use of classical methods often requires a fast surrogate for the forward model; these surrogates are frequently highly application-specific, limiting their generalizability (Tiglio & Villanueva, 2022).

Recently, generative diffusion models (Ho et al., 2020) have emerged as a powerful framework for representing expressive, multimodal priors and enabling posterior sampling in high-dimensional inverse problems (Chung et al., 2023; Li et al., 2026; Zhang et al., 2025). Since $\boldsymbol{a}$ and $\boldsymbol{u}_{\text{obs}}$ are discretized functions, a recent line of work lifts diffusion inverse solvers to infinite-dimensional function spaces (Lim et al., 2025). Existing diffusion PDE inverse solvers, including the current state-of-the-art Decoupled Diffusion Inverse Solver (DDIS) (Lin et al., 2026), first denoise a diffusion state $\boldsymbol{a}_t$ to obtain an estimate of the clean coefficient field $\hat{\boldsymbol{a}}_0$ and then separately modify this estimate to enforce measurement consistency so that $\boldsymbol{a}_0 \approx \hat{\boldsymbol{a}}_0$ and $\boldsymbol{u}_{\text{obs}} \approx \mathcal{A}(\boldsymbol{a}_0)$, where $\mathcal{A}$ is the forward operator. We refer to this class of methods as *output space optimization* methods, since this amounts to optimizing in the denoiser's output space while holding the denoiser input fixed (Figure 1a).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

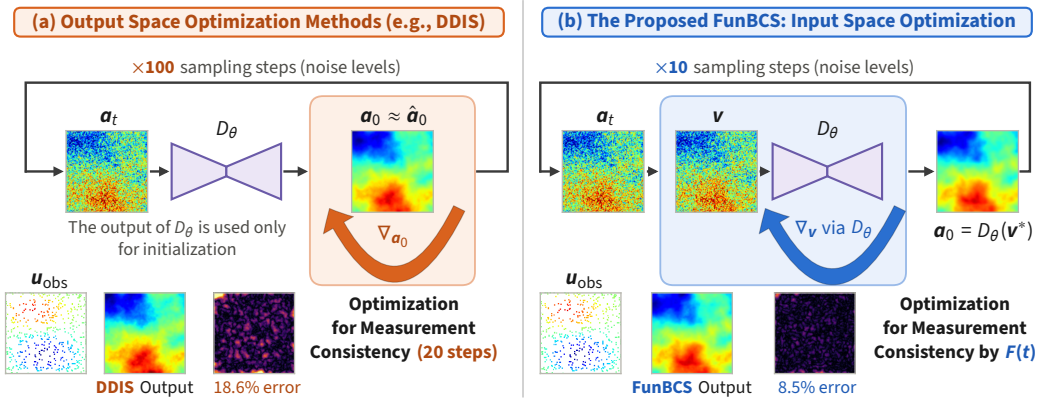


Figure 1: Overview of the differences between output space optimization methods (e.g. DDIS) (a) and our input space optimization method (b). DDIS enforces measurement consistency on a reconstruction near the denoiser's (D_θ) output, requiring 20 optimization steps at each of its 100 noise levels. In contrast, our input space optimization jointly incorporates measurement consistency and denoising by iteratively computing the denoiser output, requiring 10 noise levels for which the number of input-space optimization steps, $F(t)$, is dynamically allocated using the scheduler presented in Sec. 4.2.

However, output space optimization can be ineffective for sparsely observed PDE inverse problems. Measurement consistency updates may leave the unobserved component of the reconstruction $\boldsymbol{a}_0$ unchanged from the denoiser prediction, allowing errors in directions invisible to the forward operator to persist. This limitation is particularly pronounced for discontinuous $\boldsymbol{a}$, whose prior can exhibit distinct modes corresponding to different discontinuity locations. When these modes differ substantially in unobserved directions, measurement-consistency updates alone may fail to correct an erroneous mode selected by the denoiser. This motivates the following research question:

Can we obtain a diffusion-based PDE inverse solver that modifies both the observed and unobserved components of the denoiser prediction?

Contributions. We address the question affirmatively, showing that *input space optimization*, or selecting the *input* to the denoiser such that measurement consistency is satisfied (Fig. 1b), can effectively update unobserved components. For finite dimensional diffusion, input-space optimization is *backward-consistent* sampling (Alkhouri et al., 2025), where the reconstruction is constrained to be in the denoiser's range. Our contributions can be summarized as follows:

- **Theoretical analysis of output space optimization**. We prove that, on a discontinuous Poisson PDE inverse problem, output space optimization methods, such as DDIS, can fail to change the unobserved components of the denoiser prediction and hence fail to recover the correct prior mode.
- **Input space optimization analysis, scheduling function, and algorithm**. We theoretically demonstrate that input space optimization can change the unobserved component, thus obtaining less unobserved error than DDIS and recovering the correct prior mode. Furthermore, we analyze where to allocate most of the input optimization steps across the sampling process, and use this to design a scheduling function for various noise levels. Combining these, we introduce **Fun**ction Space **B**ackward-**C**onsistent **S**ampling (FunBCS), our diffusion input space optimization method.
- **Comprehensive evaluation**. We extensively evaluate our method across four smooth and discontinuous PDE inverse problems. Compared to the current state-of-the-art, our method reduces the reconstruction error by 27–64% (Fig. 2) while running 1.4–2.1× faster. For example, on the same hardware, FunBCS at 8 seconds per sample achieves 11.8% relative ℓ_2 error on Poisson, whereas DDIS, at best, obtains 13.7% error while running for 72 seconds (Fig. 3).

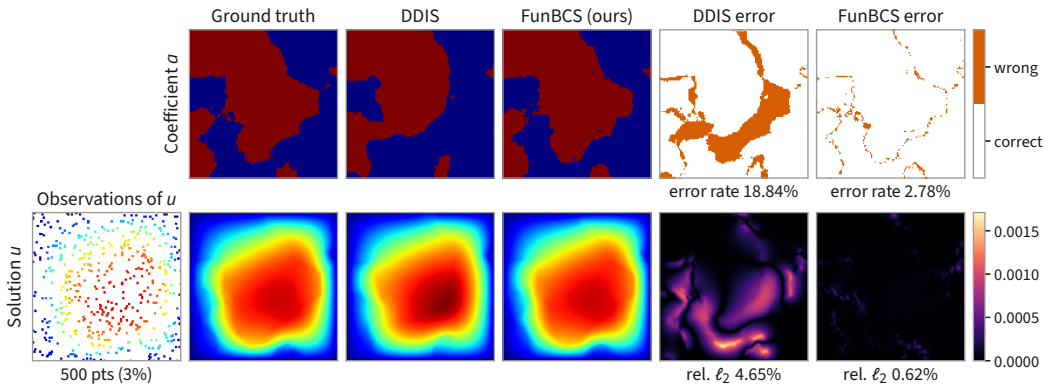


Figure 2: Darcy reconstruction from 3% sparse solution observations (first column). The last two columns show the reconstruction error. DDIS misplaces the discontinuities, mostly by assigning low permeability to regions where the true field is highly permeable, whereas FunBCS recovers the discontinuity structure significantly more faithfully.

2 Background

PDE inverse problems. A partial differential equation (PDE) can be written as:

$$\begin{cases} \mathcal{J}_{a(\cdot)} u(\cdot) = f(\cdot) & \text{on } \Omega, \\ u(\cdot) = g(\cdot) & \text{on } \partial\Omega\,, \end{cases} \tag{1}$$

for functions $u(\cdot) \in \mathcal{H}$ and $a(\cdot) \in \mathcal{F}$, where $\mathcal{H}, \mathcal{F}$ are separable Hilbert spaces, $\Omega \subset \mathbb{R}^d$ is a domain, $\partial\Omega$ is the domain boundary, and $\mathcal{J}_{a(\cdot)}$ is a (potentially nonlinear) differential operator that depends on $a(\cdot)$. The goal of a PDE inverse problem is to reconstruct physical coefficient field $a(\cdot)$ from sparse observations of PDE solution $u(\cdot)$, assuming known $f(\cdot)$ and $g(\cdot)$. Specifically, we have:

$$u_{\text{obs}}(\cdot) = \mathcal{A}(a(\cdot)) + \eta(\cdot)\,, \tag{2}$$

where $\mathcal{A}$ is a *measurement operator* and $\eta(\cdot)$ is a noise term. Letting $\mathcal{G} : \mathcal{F} \to \mathcal{H}$ be a the solution operator which maps coefficients fields to PDE solutions fields, and $\mathcal{O} : \mathcal{H} \to \mathcal{H}$ be an observation operator (e.g., a mask), we can write $\mathcal{A} = \mathcal{O} \circ \mathcal{G}$. Critically, $u_{\text{obs}}(\cdot)$ often corresponds to several possible $a(\cdot)$, and small variations in $u_{\text{obs}}(\cdot)$ result in drastically different $a(\cdot)$; that is, $\mathcal{A}$ is not injective and the inverse problem corresponding to Eq. (2) is severely ill-posed. To address this, research on PDE inverse problems often takes the Bayesian perspective (Stuart, 2010), where one assumes a prior $\pi(a(\cdot))$ on $\mathcal{F}$ and aims to sample from the posterior $\pi(a(\cdot)|u_{\text{obs}}(\cdot))$ via Bayes' theorem, i.e., $\pi(a(\cdot)|u_{\text{obs}}(\cdot)) \propto \pi(u_{\text{obs}}(\cdot)|a(\cdot))\pi(a(\cdot))$.

Throughout, we consider the *discretized* version of the problem in Eq. (2); in particular, we select n collocation points $\boldsymbol{x}_i \in \Omega$ for $i = 1, ..., n$, and evaluate only at those finite number of points[1]. For example, we discretize $a(\cdot)$ as $\boldsymbol{a} = (a(\boldsymbol{x}_1), ..., a(\boldsymbol{x}_n))$; we respectively discretize $u(\cdot), u_{\text{obs}}(\cdot), \eta(\cdot), \mathcal{G}$, and $\mathcal{O}$ as $\boldsymbol{u} \in \mathbb{R}^n$, $\boldsymbol{u}_{\text{obs}}, \boldsymbol{\eta} \in \mathbb{R}^m$, $\mathcal{G}_h : \mathbb{R}^n \to \mathbb{R}^n$, and $\mathcal{O}_h : \mathbb{R}^n \to \mathbb{R}^m$ where $m << n$. Importantly, note that $\mathcal{G}_h$ and $\mathcal{O}_h$ can remain nonlinear in this setting, and thus do not necessarily admit matrix representations. We thus let $\mathcal{A}_h = \mathcal{O}_h \mathcal{G}_h$. A detailed overview of our notation is provided in App. A.

Function space diffusion. One important way of obtaining a prior $\pi(\boldsymbol{a})$ for the PDE inverse problem in Eq. (2) is to set $\pi(\boldsymbol{a}_0) := \pi(\boldsymbol{a})$ and sample $\boldsymbol{a}_0 \sim \pi(\boldsymbol{a}_0)$ by running a learned reverse diffusion process initialized with a sample from a simple distribution $\boldsymbol{a}_T \sim \pi(\boldsymbol{a}_T)$. Since $a(\cdot)$ is a function, this involves training diffusion on (discretized) functions (Lim et al., 2025). After discretization, this is equivalent to the standard diffusion process (Ho et al., 2020) with two slight alterations, specifically that the noise has kernel matrix $\boldsymbol{K}$ as its covariance instead of $\boldsymbol{I}$ and that $\boldsymbol{K}$ multiplies the score in Tweedie's formula. More specifically, we have the variance exploding (VE) diffusion forward

[1]Without loss of generality, we note that not all traditional numerical methods use collocation points or pointwise values. For example, finite volume and finite element methods may use *averages* and spectral methods may use Fourier coefficients (Quarteroni, 2009).

process (Song et al., 2021) at $0 < t_0 < \cdots < t_N = T$, where N is the number of sampling steps:

$$\boldsymbol{a}_{t_{k+1}} = \boldsymbol{a}_{t_k} + \sqrt{\sigma_{t_{k+1}}^2 - \sigma_{t_k}^2}\, \boldsymbol{\varepsilon}_k, \qquad k = 0, \ldots, N-1, \tag{3}$$

where $\boldsymbol{\varepsilon}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{K})$ are independent across k. We perform reverse diffusion with respect to Tweedie's formula given by:

$$D^*(\boldsymbol{a}_t, t) := \mathbb{E}[\boldsymbol{a}_0 | \boldsymbol{a}_t] = \boldsymbol{a}_t + \sigma_t^2 \boldsymbol{K} \nabla_{\boldsymbol{a}_t} \log p_t(\boldsymbol{a}_t). \tag{4}$$

For a learned score $\boldsymbol{s}_{\boldsymbol{\theta}} \approx \nabla_{\boldsymbol{a}_t} \log p_t(\boldsymbol{a}_t)$ (Song et al., 2021), we denote the denoiser by $D_{\boldsymbol{\theta}} \approx \mathbb{E}[\boldsymbol{a}_0 | \boldsymbol{a}_t]$. We provide more detailed formal characterization of function space diffusion in App. B.

Function space diffusion inverse solvers. Diffusion models (Sohl-Dickstein et al., 2015) are successful at modeling high-dimensional priors (Rombach et al., 2022), including in function spaces (Lim et al., 2025). Thus, several diffusion solvers have been developed for function space inverse problems (Huang et al., 2024; Yao et al., 2025; Lin et al., 2026). At a sampling step t_k, these solvers perform two key operations: (**D**) using the diffusion denoiser to estimate the coefficient field and (**M**) enforcing measurement consistency. We use the term *output space optimization* to describe methods where (**D**) and (**M**) are *disjoint*, since this amounts to optimizing in the output space of the denoiser while leaving the denoiser input fixed. From this perspective, DiffusionPDE (Huang et al., 2024), FunDPS (Yao et al., 2025), and Decoupled Diffusion Inverse Solver (DDIS) (Lin et al., 2026) are all output space optimization procedures. We discuss these solvers in further detail in App. E. We provide an extensive related work in App. C covering classical numerical solvers, diffusion inverse solvers and theory, approaches to updating unobserved components, and guidance scheduling.

3 Analysis of Output Space Optimization Sampling

We instantiate DDIS as an example output space optimization method for which we provide a discontinuous PDE inverse problem, demonstrating that unobserved differences in prior modes arise naturally. Consequently, DDIS does not alter the denoiser's unobserved component, and hence fails to recover the correct prior mode. All proofs are deferred to App. K.

To sample from $\pi(\boldsymbol{a}|\boldsymbol{u}_{\text{obs}})$, diffusion inverse solvers, as discussed earlier, do the following at timestep t_k with noisy instance $\boldsymbol{a}_{t_k}$:

- (**D**) compute the estimated output sample $\hat{\boldsymbol{a}}_0^{(t_k)}$ via, e.g., Eq. (4) from $\boldsymbol{a}_{t_k}$ and
- (**M**) enforce *measurement consistency* on the reconstruction $\boldsymbol{a}_0^{(t_k)} \approx \hat{\boldsymbol{a}}_0^{(t_k)}$, i.e., incorporate likelihood information $\pi(\boldsymbol{u}_{\text{obs}}|\boldsymbol{a})$ so that $\boldsymbol{u}_{\text{obs}} \approx \mathcal{A}_h(\boldsymbol{a}_0^{(t_k)})$.

The defining characteristic of output space inverse solvers is that the operation **(D)** and **(M)** are performed *separately* at time t_k. Notably, output space inverse solvers may perform other operations before, between, or after **(D)** and **(M)**. Next, we formalize the DDIS sampler. Fix consecutive diffusion times $t_k > t_{k-1}$. DDIS assumes Gaussian likelihood at timestep 0, with $\gamma > 0$, yielding $p(\boldsymbol{u}_{\text{obs}} \mid \boldsymbol{a}_0) \propto \exp\left\{-\frac{\|\mathcal{A}_h \boldsymbol{a}_0 - \boldsymbol{u}_{\text{obs}}\|^2}{2\gamma^2}\right\}$. At time t_k, DDIS approximates the denoising law $\pi(\boldsymbol{a}_0 \mid \boldsymbol{a}_{t_k})$ with $\pi_D(\boldsymbol{a}_0 \mid \boldsymbol{a}_{t_k}^{(D)})$ which has density $q_D(\boldsymbol{a}_0 \mid \boldsymbol{a}_{t_k}^{(D)}) = \mathcal{N}(\boldsymbol{a}_0; \hat{\boldsymbol{a}}_0^{(t_k)}, r_{t_k}^2 \boldsymbol{I})$ with $\hat{\boldsymbol{a}}_0^{(t_k,D)} := D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)}, t_k)$, where superscript (D) denotes DDIS. Setting $r_{t_k} = \sigma_{t_k}$ gives:

$$q_{D,t_k}(\boldsymbol{a}_0 \mid \boldsymbol{a}_{t_k}^{(D)}, \boldsymbol{u}_{\text{obs}}) \propto \exp\left\{-\frac{\|\mathcal{A}_h \boldsymbol{a}_0 - \boldsymbol{u}_{\text{obs}}\|^2}{2\gamma^2} - \frac{\|\boldsymbol{a}_0 - \hat{\boldsymbol{a}}_0^{(t_k,D)}\|^2}{2\sigma_{t_k}^2}\right\}. \tag{5}$$

The next iteration of DDIS is given by:

$$\boldsymbol{a}_{t_{k-1}}^{(D)} = \tilde{\boldsymbol{a}}_0^{(t_k)} + \sigma_{t_{k-1}} \boldsymbol{\varepsilon}_k^{(D)}, \qquad \tilde{\boldsymbol{a}}_0^{(t_k)} \sim q_{D,t_k}(\cdot | \boldsymbol{a}_{t_k}^{(D)}, \boldsymbol{u}_{\text{obs}}),\ \boldsymbol{\varepsilon}_k^{(D)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{K}), \tag{6}$$

where $\varepsilon_k^{(D)}$ are drawn independently. In practice, DDIS uses Langevin dynamics (Roberts & Tweedie, 1996) to sample from Eq. (5) for use in Eq. (6) at every diffusion timestep t_k. In particular, DDIS finds $\tilde{\boldsymbol{a}}_0^{(t_k)}$ close to $\hat{\boldsymbol{a}}_0^{(t_k,D)}$, enforces measurement consistency on $\boldsymbol{a}_0$, and then renoises to t_{k-1}. This is run until time t_0 to obtain a final reconstructed physical coefficient field.

Next, towards explaining why output space optimization can fail, we provide an example PDE inverse problem on which we will show that DDIS fails:

$$u''(\cdot) = a(\cdot), \quad \text{on } (0,1), \quad u(1) = u(0) = 0. \tag{7}$$

We now assume that $a(\cdot)$ has a single discontinuity at $z \in (0,1)$, so that admissible coefficient fields are $\{a(\cdot) \in L^2(0,1) : a(x) = a^{(1)}\mathbb{1}_{(0,z)} + a^{(2)}\mathbb{1}_{[z,1]},\ z \in (0,1)\}$, where L^2 are square-integrable functions. We assume that the coefficients $a^{(1)}, a^{(2)}$ are known beforehand so that the uncertainty is only over the discontinuity location; upon reconstruction one can place $a^{(1)}, a^{(2)}$ on either side of z. Furthermore, we consider two physical coefficient fields $a_-(\cdot)$ and $a_+(\cdot)$ with respective distinct discontinuity locations $z_- \neq z_+$. With discretization at n collocation points $\boldsymbol{x}_i \in (0,1)$, consider a prior which weights these two discontinuity locations equally, i.e., $\pi(\boldsymbol{a}_0) = \frac{1}{2}\delta_{\boldsymbol{a}_-} + \frac{1}{2}\delta_{\boldsymbol{a}_+}$.

In this setting, we consider a noiseless Bayesian inverse problem per Eq. (2) given by $\boldsymbol{u}_{\text{obs}} = \boldsymbol{O}\boldsymbol{G}(\boldsymbol{a}_+)$, where $\boldsymbol{G}$ is a linear solution operator and $\boldsymbol{O}$ is a selection matrix. Thus, given $\boldsymbol{u}_{\text{obs}}$, a successful diffusion inverse solver should recover $\boldsymbol{a}_+$ at the end of the sampling process. Note that while the prior on $\boldsymbol{a}_+$ and $\boldsymbol{a}_-$ is not exactly learnable by a diffusion model, it can be approximated by a mixture of Gaussians with small variance. The noiselesss setting of $\boldsymbol{\eta} = \boldsymbol{0}$ matches both our experiments in Sec. 5 as well as the DDIS experiments in Lin et al. (2026).

Note that this setting is a linear analogue of the nonlinear Darcy inverse problem in Fig. 2 where we assume that $a^{(1)}$ and $a^{(2)}$ are known, and find the correct discontinuity location for linear Poisson. We simplify this setting by assuming only two possible discontinuity locations; we find that this is sufficient to observe the limitations of output space optimization methods, and the benefits of input space optimization (which we will discuss in the upcoming sections).

Remark 1. Let $\boldsymbol{A} := \boldsymbol{O}\boldsymbol{G}$ be the forward operator and $\boldsymbol{P}_{\ker \boldsymbol{A}}$ be the orthogonal projection matrix onto its nullspace. We prove in Prop. 8, described in App. K.11, that the setting described above yields that the measurements differ ($\boldsymbol{A}\boldsymbol{a}_+ \neq \boldsymbol{A}\boldsymbol{a}_-$) and that the nullspace components differ ($\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}_+ \neq \boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}_-$) with high probability. We thus take these as assumptions throughout.

In the setting above, we find that output space optimization can do no better than the denoiser in unobserved error. This issue is critical if the denoiser chooses the incorrect prior mode. Formally:

Proposition 1 (DDIS preserves denoiser nullspace component)**.** *Fix $t_k > t_{k-1}$. Suppose $\boldsymbol{A}\boldsymbol{a}_+ \neq \boldsymbol{A}\boldsymbol{a}_-$ and $\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}_+ \neq \boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}_-$ and let $\bar{\boldsymbol{a}}_{D,t_k-1} := \mathbb{E}[\boldsymbol{a}^{(D)}_{t_k-1} | \boldsymbol{a}^D_{t_k}, \boldsymbol{u}_{\text{obs}}]$, where $\boldsymbol{a}^{(D)}_{t_k-1}, \boldsymbol{a}^{(D)}_{t_k}$ are given by Eq.* (6) *under the exact denoiser $D^*(\cdot, t_k)$ in Eq.* (4)*. We then have that:*

$$\|\boldsymbol{P}_{\ker \boldsymbol{A}}(\bar{\boldsymbol{a}}_{D,t_k-1} - \boldsymbol{a}_+)\|_2 = \|\boldsymbol{P}_{\ker \boldsymbol{A}}(D^*(\boldsymbol{a}^{(D)}_{t_k}, t_k) - \boldsymbol{a}_+)\|_2\ . \tag{8}$$

Remark 2. Per Eq. (8), DDIS at step t_{k-1} does not grow any closer in nullspace to $\boldsymbol{a}_+$ than the denoiser does at step t_k. This is precisely because the measurement consistency error in Eq. (5) does not update the unobserved components, while the second term aligns the reconstruction with the denoiser. While the reconstruction $\tilde{\boldsymbol{a}}_0^{(t_k)}$ influences the denoiser input at the next diffusion timestep after renoising, this merely gives the denoiser another chance to correct itself, which is not guaranteed. Consequently, if the denoiser remains mistaken at later timesteps, particularly near the end of the reverse process, the corresponding nullspace error may persist in the final reconstruction. We provide experimental validation for this on Darcy flow in Sec. 5.

We note that in practice DDIS uses the probability flow ODE instead of the denoiser; per Lemma 9 in App. K.1, the above Prop. 1 still holds in this case. Prop. 1 immediately yields that DDIS fails to recover $\boldsymbol{a}_+$ more generally:

Corollary 1 (DDIS does not recover $\boldsymbol{a}_+$)**.** *Fix $t_k > t_{k-1}$. With the same assumptions as Prop. 1, let $k_{t_k}(\boldsymbol{v}) = \mathbb{P}(\boldsymbol{a}_0 = \boldsymbol{a}_+ | \boldsymbol{a}_{t_k} = \boldsymbol{v})$ be the regular conditional probability of recovering the correct $\boldsymbol{a}_+$. Then, this probability is well-defined and we have that:*

$$\|\bar{\boldsymbol{a}}_{D,t_k-1} - \boldsymbol{a}_+\|_2 \geq (1 - k_{t_k}(\boldsymbol{a}^{(D)}_{t_k}))\|\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d}\|_2 > 0. \tag{9}$$

Remark 3. Note that Cor. 1 holds for t_k arbitrary, including at the end of the sampling process, and that Eq. (9) continues to hold as $\sigma_{t_0} \to 0$. Furthermore, as discussed in Remark 2, DDIS' inability to recover the true $\boldsymbol{a}_+$ is exacerbated when the wrong mode is preferred (i.e., when $k_{t_k}(\boldsymbol{a}^{(D)}_{t_k}) < \frac{1}{2}$), particularly near end of the reverse process; we confirm this empirically on Darcy flow in Sec. 5.

Algorithm 1 Function Space Backward-Consistent Sampling (FunBCS)

Require: observations $\boldsymbol{u}_{\text{obs}}$; observation operator $\mathcal{O}$; denoiser $D_{\boldsymbol{\theta}}$; neural operator surrogate $\mathcal{G}_{h,\phi}$; diffusion schedule $\{\sigma_{t_k}\}$; optimization step function $F(t_k)$; Adam step size α.

$\boldsymbol{a}_{t_N}^{(B)} \leftarrow \sigma_{t_N}\,\varepsilon_N, \quad \varepsilon_N \sim \mathcal{N}(0, \boldsymbol{K})$ ▷ initial latent drawn from the training GRF
for $k = N, \ldots, 1$ **do**
  $\boldsymbol{v}_0 \leftarrow \boldsymbol{a}_{t_k}^{(B)}$ ▷ warm start at the current noisy state
  for $s = 0, \ldots, F(t_k) - 1$ **do**
    $\mathcal{J} \leftarrow \frac{1}{2\gamma^2}\|\mathcal{O}_h \circ \mathcal{G}_{h,\phi}(D_{\boldsymbol{\theta}}(\boldsymbol{v}_s, \sigma_{t_k})) - \boldsymbol{u}_{\text{obs}}\|_2^2$ ▷ compute denoiser output (**D**)
    $\boldsymbol{v}_{s+1} \leftarrow \text{AdamStep}_\alpha(\boldsymbol{v}_s,\, \nabla_{\boldsymbol{v}_s}\mathcal{J})$ ▷ enforce measurement consistency (**M**)
  end for
  $\boldsymbol{a}_{t_k-1}^{(B)} \leftarrow D_{\boldsymbol{\theta}}(\boldsymbol{v}_{F(t_k)}, \sigma_{t_k}) + \sigma_{t_{k-1}}\,\boldsymbol{\varepsilon}_K, \quad \boldsymbol{\varepsilon}_K \sim \mathcal{N}(0, \boldsymbol{K})$ ▷ renoise
end for
return $\boldsymbol{a}_{t_0}^{(B)}$

4 Input Space Optimization Sampling in Function Space

In this section, we propose merging steps (**D**) and (**M**) discussed above, iteratively denoising and finding the *input* to the denoiser such that measurement consistency holds. We demonstrate theoretically that this scheme does not fail in the same Poisson setting of the previous section. Subsequently, we provide insights and a procedure on how to select the number of optimization steps for the input at each sampling iteration, and present the FunBCS algorithm. To motivate input space optimization, we note the following fact about the denoiser in the setting of Sec. 3:

Proposition 2 (Denoiser nullspace component)**.** *Suppose $\boldsymbol{A}\boldsymbol{a}_+ \neq \boldsymbol{A}\boldsymbol{a}_-$ and $\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}_+ \neq \boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}_-$ hold. Let $\boldsymbol{d} = \boldsymbol{a}_+ - \boldsymbol{a}_-$ be the difference between the two prior modes, and $\boldsymbol{v}$ be a fixed denoiser input. We then have:*

$$D^*(\boldsymbol{v}, t_k) = \boldsymbol{a}_- + k_{t_k}(\boldsymbol{v})\boldsymbol{d}, \qquad \boldsymbol{P}_{\ker \boldsymbol{A}} D^*(\boldsymbol{v}, t_k) = \boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}_- + k_{t_k}(\boldsymbol{v})\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d}\,. \tag{10}$$

Per Prop. 2, the unobserved component of $D^*(\cdot, t_k)$ can be changed by updating $k_t(\boldsymbol{v})$; since $k_t(\cdot)$ is a function of the denoiser input, it suffices to optimize in the input space of the denoiser. Thus, to enable input space optimization, at every t_k, we propose solving:

$$\arg\min_{\boldsymbol{v}} \|\mathcal{O}_h \circ \mathcal{G}_{h,\boldsymbol{\phi}} \circ D_{\boldsymbol{\theta}}(\boldsymbol{v}, t_k) - \boldsymbol{u}_{\text{obs}}\|_2^2\,, \qquad \mathcal{A}_h = \mathcal{O}_h \circ \mathcal{G}_{h,\boldsymbol{\phi}}\,, \tag{11}$$

for a neural surrogate $\mathcal{G}_{h,\boldsymbol{\phi}}$, since $\mathcal{G}_h$ may be difficult to differentiate. Using superscript (B) for FunBCS, consider the gradient flow on $\boldsymbol{v}$ with respect to the objective in Eq. (11) with $\gamma > 0$ and optimization time $s \geq 0$:

$$\frac{d\boldsymbol{v}(s)}{ds} = -\frac{1}{2\gamma^2}\,\nabla_{\boldsymbol{v}}\|\mathcal{O}_h \circ \mathcal{G}_{h,\boldsymbol{\phi}} \circ D_{\boldsymbol{\theta}}(\boldsymbol{v}, t_k) - \boldsymbol{u}_{\text{obs}}\|^2\big|_{\boldsymbol{v}=\boldsymbol{v}(s)}\,, \qquad \boldsymbol{v}(0) = a_{t_k}^{(B)}\,. \tag{12}$$

Remark 4. After discretizing Eq. (12), this amounts to iteratively denoising and optimizing the input, thus merging steps (**D**) and (**M**). Under the exact denoiser, the solution of Eq. (12) exists and is unique per Lemma 6 in App. K.1, and the probability $k_{t_k}(\boldsymbol{v}(s))$ is monotonically increasing in s per Lemma 7 in App. K.1. Additionally, input space optimization is equivalent to *preconditioned* output space optimization, with the preconditioner $\nabla_{\boldsymbol{v}} D^*(\boldsymbol{v}, t_k)\nabla_{\boldsymbol{v}} D^*(\boldsymbol{v}, t_k)^T|_{\boldsymbol{v}=\boldsymbol{v}(s)}$; we discuss this further in App. J.

Solving Eq. (11) with Eq. (12) and renoising yields the update rule at a diffusion time t_k and optimization time s, for $\boldsymbol{\varepsilon}_k^{(B)}$ drawn independently:

$$\boldsymbol{a}_{t_{k-1}}^{(B)} = D_{\boldsymbol{\theta}}(\boldsymbol{v}(s), t_k) + \sigma_{t_{k-1}}\boldsymbol{\varepsilon}_k^{(B)}, \qquad \boldsymbol{\varepsilon}_k^{(B)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K})\,. \tag{13}$$

4.1 Analysis of Input Space Optimization

To analyze input space optimization, we revisit the setting of Sec. 3, where we assume access to the exact operator $\boldsymbol{G}$. We begin by analyzing how the recurrence in Eq. (13) changes the nullspace error:

Proposition 3 (Input space optimization reduces nullspace error better)**.** *Under the same assumptions as Prop. 1, we have that* $\mathbb{E}[\boldsymbol{a}_{t_k-1}^{(B)}|\boldsymbol{a}_{t_k}^{(B)}, \boldsymbol{u}_{\text{obs}}] = D^*(\boldsymbol{v}(s), t_k)$ *and for any* $s > 0$*:*

$$\|\boldsymbol{P}_{\ker \boldsymbol{A}}(D^*(\boldsymbol{v}(s), t_k) - \boldsymbol{a}_+)\|_2 \leq \|\boldsymbol{P}_{\ker \boldsymbol{A}}(D^*(\boldsymbol{a}_{t_k}^{(B)}, t_k) - \boldsymbol{a}_+)\|_2 \,. \tag{14}$$

Additionally, there exists an $s^* > 0$ *such that* $s \geq s^*$ *yields:*

$$\|\boldsymbol{P}_{\ker \boldsymbol{A}}(D^*(\boldsymbol{v}(s), t_k) - \boldsymbol{a}_+)\|_2 \leq \|\boldsymbol{P}_{\ker \boldsymbol{A}}(\bar{\boldsymbol{a}}_{D,t_k-1} - \boldsymbol{a}_+)\|_2 \,. \tag{15}$$

Remark 5. As expected per our discussion in Prop. 2, the input space optimization recurrence in Eq. (13) contracts nullspace error better than both the denoiser evaluated at the previous iterate and DDIS, since it can update $k_t(\boldsymbol{v})$. We demonstrate Prop. 3 empirically in App. I.3.

Furthermore, input space optimization *can* recover the correct $\boldsymbol{a}_+$ in this PDE inverse setting:

Theorem 1 (Input space optimization can recover $\boldsymbol{a}_+$)**.** *Consider diffusion times* $t_k > t_{k-1}$*. Under the same assumptions as Prop. 1, we have that:*

$$\pi_s(\boldsymbol{a}_{t_k-1}^{(B)}|\boldsymbol{a}_{t_k}^{(B)}, \boldsymbol{u}_{\text{obs}}) = \mathcal{N}(D^*(\boldsymbol{v}(s), t_k), \sigma_{t_k-1}^2 \boldsymbol{K}) \Rightarrow \mathcal{N}(\boldsymbol{a}_+, \sigma_{t_k-1}^2 \boldsymbol{K}), \tag{16}$$

as $s \to \infty$*. At the final step* $t_{k-1} = t_0$ *the limiting law converges in distribution to* $\delta_{\boldsymbol{a}_+}$ *as* $\sigma_{t_0} \to 0$*.*

Thus, for a multimodal prior and with differences between prior modes partially lying in measurement nullspace, which arises naturally in the discontinuous PDE inverse problem setting per Prop. 8 in App. K.11, input space optimization can significantly outperform output optimization. We note that this happens regardless if the denoiser begins by preferring the wrong mode; we provide experimental validation for this on Darcy flow in Sec. 5. The resulting algorithm not only provides a mechanism to change the unobserved components of the denoiser prediction (as we theoretically support), but also reduces the need of a large number of sampling steps as we show in our experimental results Sec. 5.

4.2 Allocating Input Space Optimization Steps Across Noise Levels

In this subsection, we address the question of how many measurement consistency steps we should take per one noise level (or sampling step). Let $F(t_k)$ denote the number of gradient flow steps taken at noise level t_k, so that the sampler takes $M = \sum_{k=1}^{N} F(t_k)$ steps in total. Every such step differentiates through the denoiser, so this allocation sets the cost of the sampler.

To make our sampler more efficient, we ask how a fixed budget of steps should be distributed across noise levels, and we answer in the setting of Sec. 3. We begin with the case in which the sampler leans towards the wrong prior mode and a noise level receives no optimization steps at all. In particular, for $t_{k+1} > t_k$, we let $k_{t_{k+1}}(\boldsymbol{a}_{t_{k+1}}^{(B)}) \leq \frac{1}{2} - \delta$ for $\delta \in (0, 1/2)$ and $\boldsymbol{a}_{t_k}^{(B)} = D^*(\boldsymbol{a}_{t_{k+1}}^{(B)}, t_{k+1}) + \sigma_{t_k}\boldsymbol{\varepsilon}_k^{(B)}$. We can then characterize the number of gradient flow steps necessary to escape to the correct mode in the subsequent diffusion step t_k as follows:

Proposition 4 (Time explosion with no optimization steps)**.** *Consider the setting of Sec. 4.2 with* $\boldsymbol{A}\boldsymbol{a}_+ \neq \boldsymbol{A}\boldsymbol{a}_-$ *and* $\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}_+ \neq \boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}_-$*. Let* $\boldsymbol{v}_{t_k}(s)$ *solve Eq.* (12) *and define the hitting time* $\tau_{t_k} := \inf\left\{s \geq 0 : k_{t_k}(\boldsymbol{v}_{t_k}(s)) \geq \frac{1}{2}\right\}$*. Then, for constants* $Q, C > 0$*, we have that:*

$$\mathbb{P}\left(\tau_{t_k} \geq \frac{\sigma_{t_k}^4}{C}\left[\exp\left(\frac{Q\delta}{2\sigma_{t_k}^2}\right) - 1\right] \,\middle|\, \boldsymbol{a}_{t_{k+1}}^{(B)}\right) \geq 1 - \exp\left(-\frac{Q\delta^2}{8\sigma_{t_k}^2}\right) \,. \tag{17}$$

Remark 6. For fixed $\delta > 0$, the lower bound on the hitting time tends to ∞ exponentially, with probability tending to 1 as $\sigma_{t_k} \to 0$. That is, a lean towards the wrong mode that is not corrected with input space optimization steps can result in an exponentially larger cost to correct in later steps. We conclude that we should retain a positive number of optimization steps to avoid this critical scenario.

We next consider at which noise level(s) to allocate most of the gradient flow steps in our method. To address this, we allow the noise standard deviation $\sigma > 0$ to vary continuously and consequently use σ, in place of t_k, to index. We then fix a single denoiser input leaning to the wrong mode, and see where the minimal time to escape to the correct mode is. Formally, we fix an input $\boldsymbol{v}_* \in \mathbb{R}^n$ satisfying $k_\sigma(\boldsymbol{v}_*) < \frac{1}{2}$, and compute the fastest hitting time across σ:

Table 1: Recovery of physical coefficient fields $\boldsymbol{a}$ with relative ℓ_2 error and spectral energy error E_s on 100 sampled instances from 3% observations. s/sample is seconds per sample on one NVIDIA A100. FunBCS-AC and FunBCS-EF denote FunBCS in accuracy mode and in efficiency mode, respectively (App. G). FunBCS (uniform) is FunBCS without the scheduling function, which corresponds to the case assigning $F(t_k) = 80$ at every noise level ($M = 800$). Errors are mean ± standard deviation over these instances. Lower is better.

	Poisson		Helmholtz		Navier–Stokes		Darcy		
Method	rel. ℓ_2	E_s	rel. ℓ_2	E_s	rel. ℓ_2	E_s	err. rate	E_s	s/sample
DiffusionPDE	21.20±6.86	0.150±0.119	18.93±5.04	0.133±0.084	9.49±1.65	0.230±0.025	7.49±4.60	0.311±0.169	274.5
FunDPS	22.05±7.11	0.169±0.116	16.25±4.09	0.156±0.077	8.18±1.53	0.381±0.032	5.04±2.71	0.152±0.060	14.5
DDIS	15.76±5.08	0.071±0.053	15.47±4.34	0.070±0.051	8.36±1.55	**0.158**±0.017	7.48±3.89	0.165±0.089	18.0
FunBCS (uniform)	10.51±3.38	0.050±0.038	9.90±2.71	**0.056**±0.034	6.23±1.13	0.300±0.069	2.82±1.56	0.113±0.053	16.5
FunBCS-EF	10.22±3.33	0.056±0.049	10.04±2.95	0.085±0.053	6.14±1.11	0.280±0.059	2.69±1.50	0.086±0.041	**10.4**
FunBCS-AC	**9.60**±3.08	**0.047**±0.039	**9.06**±2.51	0.069±0.037	**5.89**±1.04	0.288±0.051	**2.32**±1.27	**0.080**±0.033	16.5

Proposition 5 (Selection window)**.** *Under the assumptions of Prop. 4, let $w_\sigma(s)$ denote $k_\sigma(\boldsymbol{v}(s))$ for $\boldsymbol{v}(s)$ a solution to Eq.* (12) *at noise level σ with initialization $\boldsymbol{v}_*$, and define $\tau(\sigma) := \inf\{s \geq 0 : w_\sigma(s) \geq \frac{1}{2}\}$. Then, there exist $\sigma_{\text{low}}, \sigma_{\text{high}} \in (0, \infty)$ and $\sigma^\star \in [\sigma_{\text{low}}, \sigma_{\text{high}}]$ such that $\tau(\sigma^\star) = \min_{\sigma>0} \tau(\sigma)$.*

Remark 7. At very high noise levels, the denoiser is undecided between the two modes, but its output responds only weakly to its input, so the gradient flow moves prohibitively slowly. At very low noise levels, the denoiser responds strongly to input changes, but it has already committed to the wrong mode. The correct prior mode is therefore selected most cheaply at intermediate noise levels, inside an interval $[\sigma_{\text{low}}, \sigma_{\text{high}}]$ that we call the *selection window*. By our experiments in Sec. 5 with our scheduler described in App. G, we find this generalizes across PDE.

Together, the two propositions constrain how a budget of optimization steps should be distributed. By Prop. 4, every noise level must retain a positive number of steps, so that a lean towards the wrong mode is never carried uncorrected into the lower noise levels. By Prop. 5, the remaining steps are best spent inside the selection window. We therefore treat the total number of optimization steps M as a budget, which a scheduling function allocates across noise levels by choosing $F(t_k)$, which keeps a floor of steps at every level and concentrates the rest in a computed selection window, with further details in App. G. At a fixed budget, this moves steps to where they matter; at a smaller budget, it removes the steps that matter least.

Algorithm. Our sampler is given in Algorithm 1. Equation (13) yields backward-consistency sampling, but not in the finite-dimensional regime as in Alkhouri et al. (2025). This means that the reconstruction $\boldsymbol{a}_0^{(t_k)}$ before renoising is *exactly* the output of the denoiser, rather than approximately.

5 Experiments

Setup. We evaluate on four PDE inverse problems, each at a resolution of 128^2 points: *Poisson*, *Helmholtz*, *Navier-Stokes* (NS) and *Darcy flow*, each observed at 500 randomly placed points of the solution field (3% of the grid) without noise following Lin et al. (2026). We report the relative ℓ_2 error of the recovered coefficient field in percent and the spectral energy error E_s averaged across 100 inference samples. FunBCS uses $N = 10$ noise levels (sampling steps) with two scheduling schemes, Accuracy mode (AC) and Efficiency mode (EF). See App. G for details. Other experimental details are given in App. F.

Main reconstruction results. Per Table 1, our sampler attains the lowest reconstruction error on every task when compared to baselines. Relative to DDIS, the error falls by 39% on Poisson, 41% on Helmholtz, 30% on Navier-Stokes, and 69% on Darcy. Box plots of the Table 1 errors are visualized in Fig. 15 in App. I.2. In line with our theory, the largest gain is on Darcy flow, whose piecewise-constant coefficient makes the prior sharply multimodal: the error rate drops from 7.48% to 2.32%, a 3.2× reduction, and the visualization in Fig. 2 show that DDIS misplaces interfaces whereas our reconstructions recover the correct discontinuity locations and geometries. Qualitative visualizations of the results on all four PDEs are in Figs. 16 and 17 in App. I.1. The spectral energy error of our method is better on Poisson, Helmholtz and Darcy, whereas DDIS retains a lower E_s on Navier-Stokes despite its higher ℓ_2 error. This is because E_s only checks whether a reconstruction

has the right amount of detail at each spatial scale, not whether that detail is in the right place. As we show in App. H, DDIS produces more fine-scale detail than FunBCS, which brings this amount closer to that of the ground truth, but neither method places the fine-scale detail where it appears in the true field.

Impact of the proposed scheduling function. At a budget of $M = 800$ steps, concentrating them in the selection window (i.e., FunBCS-AC) lowers the error when compared to the uniform allocation (i.e., FunBCS without the scheduling function in Table 1). For example, on Poisson, the error is reduced by 9%. Trimming the steps outside the window, i.e., FunBCS-EF, removes 24–49% of them, yet FunBCS-EF still outperforms every baseline on every task, running 1.4–$2.1\times$ faster than DDIS with 27–64% lower reconstruction error (App. G.1). We show in App. G.3 that our scheduling function also improves DDIS on its efficiency, though not as much in magnitude as the gain we observed for FunBCS.

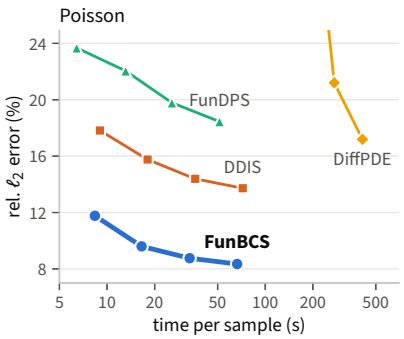


Figure 3: Poisson reconstruction error with respect to the time per sample. All methods are run using the same hardware. FunBCS and DDIS use the same diffusion prior.

Scaling with compute. Figure 3 presents how the Poisson error of every method scales (Fig. 6 in App. F present the results on the other PDEs). At every time budget, FunBCS is more accurate than baselines. For example, FunBCS at 8 seconds per sample achieves 11.8% relative ℓ_2 error on Poisson, whereas DDIS, at best, obtains 13.7% error while running for 72 seconds.

FunBCS ability of escaping a wrong mode. In Sec. 3 and Sec. 4, we highlighted that DDIS on a discontinuous Poisson inverse problem can fail when the denoiser selects the wrong mode, while FunBCS fixes this problem. We test this mechanism on Darcy flow, where several possible discontinuity locations result in a learned multimodal prior and, as discussed in Sec. 3 and Prop. 8 in App. K.11, a component of the denoiser decision lies in measurement nullspace with high probability. Results across 50 test instances are given in Fig. 4 where we move every discontinuity location of the true field $\boldsymbol{a}_+$ by five pixels to obtain a wrong field $\boldsymbol{a}_-$, resulting in 11% pixel error. We find that when started at $\sigma = 0.5$, FunBCS escapes $\boldsymbol{a}_-$ on 94% of the instances, whereas DDIS *escapes on none*; doubling the inference budget of DDIS only increases this to 18%. We visualize this across noise levels σ in Fig. 4. This is precisely the behavior predicted by Cor. 1 for DDIS and Thm. 1 for FunBCS: if the denoiser is wrong, DDIS can fail, and this is exacerbated towards the end of the sampling process, while in contrast FunBCS can adjust the denoiser's decision and solve this problem.

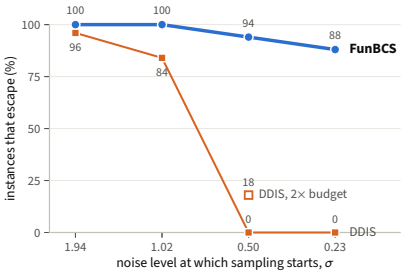


Figure 4: Share of the 50 Darcy test instances on which each sampler escapes the wrong field $\boldsymbol{a}_-$, against the noise level σ at which sampling starts from renoised $\boldsymbol{a}_-$. An instance escapes when its final reconstruction assigns most of the pixels where $\boldsymbol{a}_+$ and $\boldsymbol{a}_-$ differ to the true field $\boldsymbol{a}_+$, with a lower error rate than $\boldsymbol{a}_-$. The hollow marker is DDIS with twice its inference budget.

6 Conclusion

We introduce FunBCS, an input space optimization approach for solving sparse PDE inverse problems. We demonstrate that existing methods, such as the state-of-the-art DDIS, perform output space optimization and consequently can fail to obtain the correct coefficient field, especially when the prior is naturally multimodal (e.g., when discontinuities are present in the coefficient field). We demonstrate that FunBCS does not suffer from this issue: across four PDE inverse problems, it significantly reduces the reconstruction error of the state-of-the-art at significantly lower runtime. We discuss limitations and future directions in App. D.

AI USE STATEMENT

In this work, we used generative AI tools for verifying the proofs of our theorems and compiling our experimental data into plots. We have not used generative AI tools for proving the mathematical statements; evaluating our baselines, which are taken from the original codebases; or implementing our core method. We have reviewed all AI-assisted work, verifying proofs ourselves as well as verifying experimental results. We take responsibility for the final content of this work, including any text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide code to reproduce our experiments at `https://anonymous.4open.science/r/FunBCS-7C24/`, alongside comprehensive experimental details in App. F. We detail our FunBCS algorithm explicitly in algorithm 1, as well as the procedure used to locate the window $[\sigma_{\text{low}}, \sigma_{\text{high}}]$ in algorithm 2 and how we allocate scheduling steps in App. G. We additionally provide all proofs of our mathematical statements throughout the paper in App. K.

REFERENCES

Giovanni S Alberti and Matteo Santacesaria. Infinite dimensional compressed sensing from anisotropic measurements and applications to inverse problems in pde. *Applied and Computational Harmonic Analysis*, 2021.

Ismail R. Alkhouri, Shijun Liang, Cheng-Han Huang, Jimmy Dai, Qing Qu, Saiprasad Ravishankar, and Rongrong Wang. SITCOM: Step-wise triple-consistent diffusion sampling for inverse problems. In *International Conference on Machine Learning (ICML)*, 2025.

Martin Andrae, Erik Wikingsson, So Takao, Tomas Landelius, and Fredrik Lindsten. DAISI: Data assimilation with inverse sampling using stochastic interpolants. In *International Conference on Machine Learning (ICML)*, 2026.

Simon R Arridge. Optical tomography in medical imaging. *Inverse Problems*, 1999.

Muhammad H Ashiq, Samanyu Arora, Abhinav N Harish, Ishaan Kharbanda, Hung Y Tseng, and Grigorios G Chrysos. Why DDIM hallucinates more than DDPM: A theoretical analysis of reverse dynamics. In *International Conference on Machine Learning (ICML)*, 2026.

Stéphane Avril, Marc Bonnet, Anne-Sophie Bretelle, Michel Grédiac, François Hild, Patrick Ienny, Félix Latourte, Didier Lemosse, Stéphane Pagano, Emmanuel Pagnacco, and Fabrice Pierron. Overview of identification methods of mechanical parameters based on full-field measurements. *Experimental Mechanics*, 2008.

Junseo Bang, Dong Ju Mun, Hoigi Seo, Seongmin Hong, and Se Young Chun. Triadic dynamics aware diffusion posterior sampling for inverse problems: Optimizing guidance and stochasticity schedules. In *International Conference on Machine Learning (ICML)*, 2026.

Johnathan M Bardsley, Antti Solonen, Heikki Haario, and Marko Laine. Randomize-then-optimize: A method for sampling from posterior distributions in nonlinear inverse problems. *SIAM Journal on Scientific Computing*, 2014.

Jan-Hendrik Bastek, WaiChing Sun, and Dennis Kochmann. Physics-informed diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025.

Yurii Ya Belov. *Inverse problems for partial differential equations*. Walter de Gruyter, 2012.

Marc Bonnet and Andrei Constantinescu. Inverse problems in elasticity. *Inverse Problems*, 2005.

Benjamin Boys, Mark Girolami, Jakiw Pidstrigach, Sebastian Reich, Alan Mosca, and Omer Deniz Akyildiz. Tweedie moment projected diffusions for inverse problems. *Transactions on Machine Learning Research (TMLR)*, 2024.

Tan Bui-Thanh, Omar Ghattas, James Martin, and Georg Stadler. A computational framework for infinite-dimensional bayesian inverse problems part i: The linearized case, with application to global seismic inversion. *SIAM Journal on Scientific Computing*, 2013.

Benjamin A. Burns and Sara Fridovich-Keil. When, why, and how do diffusion posterior samplers fail? a finite-sample lens. *arXiv preprint arXiv:2605.30330*, 2026.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 2018.

Siyi Chen, Yixuan Jia, Qing Qu, He Sun, and Jeffrey Fessler. Flowdas: A stochastic interpolant-based framework for data assimilation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Nelson Christensen and Renate Meyer. Parameter estimation with gravitational waves. *Reviews of Modern Physics*, 2022.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations (ICLR)*, 2023.

Jonathan Citrin, Ian Goodfellow, Akhil Raju, Jeremy Chen, Jonas Degrave, Craig Donner, Federico Felici, Philippe Hamel, Andrea Huber, Dmitry Nikulin, David Pfau, Brendan Tracey, Martin Riedmiller, and Pushmeet Kohli. Torax: a fast and differentiable tokamak transport simulator in jax. *arXiv preprint arXiv:2406.06718*, 2024.

Simon L Cotter, Gareth O Roberts, Andrew M Stuart, and David White. Mcmc methods for functions: modifying old algorithms to make them faster. *Statistical Science*, 2013.

Evan Scope Crafts and Umberto Villa. Benchmarking diffusion annealing-based bayesian inverse problem solvers. *IEEE Open Journal of Signal Processing*, 2025.

Masoumeh Dashti, Stephen Harris, and Andrew Stuart. Besov priors for bayesian inverse problems. *Inverse Problems and Imaging*, 2012.

Matias G Delgadino, Sebastien Motsch, Advait Parulekar, William Porteous, and Sanjay Shakkottai. Diffusion-based posterior sampling: A feynman-kac analysis of bias and stability. *arXiv preprint arXiv:2605.06538*, 2026.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 2011.

Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*. Springer Dordrecht, 1996.

Oliver G Ernst, Björn Sprungk, and Hans-Jörg Starkloff. Analysis of the ensemble and polynomial chaos kalman filters in bayesian inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 2015.

Rong Ge, Holden Lee, and Andrej Risteski. Beyond log-concavity: Provable guarantees for sampling multi-modal distributions using simulated tempering langevin monte carlo. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Alex A Gorodetsky, John D Jakeman, Gianluca Geraci, and Michael S Eldred. Mfnets: Multi-fidelity data-driven networks for bayesian learning and prediction. *International Journal for Uncertainty Quantification*, 2020.

Yongtao Guan and Stephen M Krone. Small-world mcmc and convergence to multi-modal distributions: From slow mixing to fast mixing. *The Annals of Applied Probability*, 2007.

Martin Hairer, Andrew M Stuart, and Sebastian J Vollmer. Spectral gaps for a metropolis–hastings algorithm in infinite dimensions. *The Annals of Applied Probability*, 2014.

Liav Hen, Tom Tirer, Raja Giryes, and Shady Abu-Hussein. Jacobian-aware posterior sampling for inverse problems. *Transactions on Machine Learning Research (TMLR)*, 2026.

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

Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. DiffusionPDE: Generative PDE-solving under partial observation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Marco A Iglesias, Kody JH Law, and Andrew M Stuart. Ensemble kalman methods for inverse problems. *Inverse Problems*, 2013.

Marco A Iglesias, Yulong Lu, and Andrew M Stuart. A bayesian level set method for geometric inverse problems. *Interfaces and free boundaries*, 2016.

Yixuan Jia, Siyi Chen, Yida Pan, Xiao Li, Lianghe Shi, Chanyong Jung, Haijie Yuan, Ismail Alkhouri, Yue Cynthia Wu, Saiprasad Ravishankar, Jeffrey A Fessler, and Qing Qu. Forcingdas: Unified and robust data assimilation via diffusion forcing. *arXiv preprint arXiv:2605.14285*, 2026.

Yuchen Jiao, Na Li, Changxiao Cai, Yuxin Chen, and Gen Li. Provable diffusion-based posterior sampling for linear inverse problems via ddim. *arXiv preprint arXiv:2607.19333*, 2026.

George E Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 2021.

K Kashinath, M Mustafa, A Albert, J-L Wu, C Jiang, S Esmaeilzadeh, K Azizzadenesheli, R Wang, A Chattopadhyay, A Singh, et al. Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A*, 2021.

Gavin Kerrigan, Justin Ley, and Padhraic Smyth. Diffusion generative models in infinite dimensions. In *International Conference on Artificial Intelligence and Statistics*, 2023.

Peter E Kloeden and Eckhard Platen. *Numerical solution of stochastic differential equations*. Springer Berlin, Heidelberg, 1992.

Oscar Leong and Yann Traonmilin. A recovery theory for diffusion priors: Deterministic analysis of the implicit prior algorithm. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2026.

Xiang Li, Soo Min Kwon, Shijun Liang, Ismail R Alkhouri, Saiprasad Ravishankar, and Qing Qu. DCDP: Decoupled data consistency via diffusion purification for inverse problems. In *2026 IEEE International Conference on Computational Photography (ICCP)*, 2026.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations (ICLR)*, 2021.

Jae Hyun Lim, Nikola B Kovachki, Ricardo Baptista, Christopher Beckham, Kamyar Azizzadenesheli, Jean Kossaifi, Vikram Voleti, Jiaming Song, Karsten Kreis, Jan Kautz, Christopher Pal, Arash Vahdat, and Anima Anandkumar. Score-based diffusion models in function space. *Journal of Machine Learning Research (JMLR)*, 2025.

Thomas YL Lin, Jiachen Yao, Lufang Chiang, Julius Berner, and Anima Anandkumar. Decoupled diffusion sampling for inverse problems on function spaces. *International Conference on Learning Representations Workshops*, 2026.

Haibo Liu and Guang Lin. Dilo: Decoupling generative priors and neural operators via diffusion latent optimization for inverse problems. *arXiv preprint arXiv:2604.11375*, 2026.

Pinelopi Papalampidi, Olivia Wiles, Ira Ktena, Aleksandar Shtedritski, Emanuele Bugliarello, Ivana Kajić, Isabela Albuquerque, and Aida Nematzadeh. Dynamic classifier-free diffusion guidance via online feedback. In *International Conference on Learning Representations (ICLR)*, 2026.

Sungwon Park, Anthony Zhou, Hongjoong Kim, and Amir Barati Farimani. Generative neural operators through diffusion last layer. In *International Conference on Machine Learning (ICML)*, 2026.

Yinglei Peng, Marko Ristić, Atul Kedia, Richard O'Shaughnessy, Christopher J. Fontes, Chris L. Fryer, Oleg Korobkin, Matthew R. Mumpower, V. Ashley Villar, and Ryan T. Wollaeger. Kilonova light-curve interpolation with neural networks. *Physical Review Research*, 2024.

Jakiw Pidstrigach, Youssef Marzouk, Sebastian Reich, and Sven Wang. Infinite-dimensional diffusion models. *Journal of Machine Learning Research*, 2024.

Joemini Poudel, Yang Lou, and Mark A Anastasio. A survey of computational frameworks for solving the acoustic inverse problem in three-dimensional photoacoustic computed tomography. *Physics in Medicine & Biology*, 2019.

Yilan Qin, Jiayu Ma, Mingle Jiang, Chuanfei Dong, Haiyang Fu, Liang Wang, Wenjie Cheng, and Yaqiu Jin. Data-driven modeling of landau damping by physics-informed neural networks. *Physical Review Research*, 2023.

Alfio Quarteroni. *Numerical models for differential problems*. Springer, 2009.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 2019.

Gareth O Roberts and Richard L Tweedie. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 1996.

Gareth O Roberts, Andrew Gelman, and Walter R Gilks. Weak convergence and optimal scaling of random walk metropolis algorithms. *The Annals of Applied Probability*, 1997.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

François Rozet, Gérôme Andry, François Lanusse, and Gilles Louppe. Learning diffusion priors from observations by expectation maximization. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Facundo Sapienza, Jordi Bolibar, Frank Schäfer, Brian Groenke, Avik Pal, Victor Boussange, Patrick Heimbach, Giles Hooker, Fernando Pérez, Per-Olof Persson, et al. Differentiable programming for differential equations: A review. *arXiv preprint arXiv:2406.09699*, 2024.

Thomas Savary, François Rozet, and Gilles Louppe. Training-free bayesian filtering with generative emulators. In *International Conference on Machine Learning (ICML)*, 2026.

Siming Shan, Min Zhu, Youzuo Lin, and Lu Lu. Regularization by denoising diffusion models for solving inverse pde problems with application to full waveform inversion. *Communications Physics*, 2026.

Xiao Shi, Edward P. Chandler, Chicago Y. Park, Shirin Shoushtari, and Ulugbek S. Kamilov. Nullflow: One-step generative reconstruction. *arXiv preprint arXiv:2606.22696*, 2026a.

Yaozhong Shi, Zachary E Ross, Domniki Asimaki, and Kamyar Azizzadenesheli. Stochastic process learning via operator flow matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Yaozhong Shi, Zachary E. Ross, and Yisong Yue. Flow annealing posterior sampling for function-space regression and inverse problems. *arXiv preprint arXiv:2606.22346*, 2026b.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning (ICML)*, 2015.

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

Andrew M Stuart. Inverse problems: a bayesian perspective. *Acta Numerica*, 2010.

Llewellyn Hilleth Thomas. Elliptic problems in linear difference equations over a network. *Watson Sci. Comput. Lab. Rept., Columbia University, New York*, 1949.

Manuel Tiglio and Aarón Villanueva. Reduced order and surrogate models for gravitational waves. *Living Reviews in Relativity*, 2022.

Utkarsh Utkarsh, Pengfei Cai, Alan Edelman, Rafael Gomez-Bombarelli, and Christopher Vincent Rackauckas. Physics-constrained flow matching: Sampling generative models with hard constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Jean Virieux and Stéphane Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 2009.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *International Conference on Learning Representations (ICLR)*, 2023.

Yuchen Wu, Minshuo Chen, Zihao Li, Mengdi Wang, and Yuting Wei. Theoretical insights for diffusion guidance: A case study for gaussian mixture models. In *International Conference on Machine Learning (ICML)*, 2024a.

Zihui Wu, Yu Sun, Yifan Chen, Bingliang Zhang, Yisong Yue, and Katherine L Bouman. Principled probabilistic imaging using diffusion models as plug-and-play priors. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b.

Tongda Xu, Xiyan Cai, Xinjie Zhang, Xingtong Ge, Dailan He, Ming Sun, Jingjing Liu, Ya-Qin Zhang, Jian Li, and Yan Wang. Rethinking diffusion posterior sampling: From conditional score estimator to maximizing a posterior. In *International Conference on Learning Representations (ICLR)*, 2025.

Yiming Yang, Xiaoyuan Cheng, Yi He, Kaiyu Li, Wenxuan Yuan, and Zhuo Sun. Outlier-robust diffusion posterior sampling for bayesian inverse problems. *arXiv preprint arXiv:2602.02045*, 2026.

Jiachen Yao, Abbas Mammadov, Julius Berner, Gavin Kerrigan, Jong Chul Ye, Kamyar Azizzadenesheli, and Anima Anandkumar. Guided diffusion sampling on function spaces with applications to PDEs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Nebiyou Yismaw, Ulugbek S Kamilov, and M Salman Asif. Gaussian is all you need: A unified framework for solving inverse problems via diffusion posterior sampling. *IEEE Transactions on Computational Imaging*, 2025.

Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Appendix

Table of Contents

A Notation **16**

B Function Space Diffusion Setup **16**

C Extended Related Work **16**
C.1 Classical Inverse Solvers 16
C.2 Physics-Informed Learning 17
C.3 Theoretical Guarantees 19

D Limitations & Future Directions **20**

E Overview of Recent Diffusion PDE Inverse Solvers **20**

F Experimental Details **21**

G The Proposed Scheduling Function **23**
G.1 Scheduler Ablations 24
G.2 Tuning the Selection Window 26
G.3 Generalizing Scheduling Function to DDIS 27

H Navier–Stokes Spectral Analysis **27**

I Additional Experiments **28**
I.1 Visualizing Reconstructed Fields & Solutions 28
I.2 Box Plots of Main Results 28
I.3 Study of Discontinous Poisson's Equation 29

J Connection with Preconditioned Output Space Optimization **29**

K Proofs **32**
K.1 Helpful Lemmas 32
K.2 Proof of Proposition 1 40
K.3 Proof of Proposition 2 41
K.4 Proof of Corollary 1 41
K.5 Proof of Proposition 3 41
K.6 Proof of Theorem 1 42
K.7 Proof of Proposition 4 43
K.8 Proof of Proposition 5 44
K.9 Proof of Proposition 6 46
K.10 Proof of Proposition 7 46
K.11 Statement and Proof of Proposition 8 48

A NOTATION

We use $a(\cdot)$ to denote a function; $\boldsymbol{a}$ to denote a vector, which may be a discretization of function $a(\cdot)$ at collocation points $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$; and a to denote a scalar. We use $\mathcal{A}$ to denote an operator, $\mathcal{A}_h$ to denote a discretized operator at points $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$; and $\boldsymbol{A}$ to denote a linear operator (i.e., a matrix). We use π to denote a probability distribution and p or q to denote its density, if it exists. $\nabla_{\boldsymbol{x}} f(\boldsymbol{x})$ is taken to be either the gradient or Jacobian of f, depending on the codomain dimension of f. We use $\mathbb{1}_A$ to denote the indicator on event A. We use $\boldsymbol{a}_{t_k}$ to denote a sample drawn from VE diffusion reverse process timestep t_k, per Eq. (20), and $\boldsymbol{a}_0$ to denote a sample drawn from the end of this process; $\hat{\boldsymbol{a}}_0^{(t_k)} = D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}, t_k)$ to denote the denoiser output given sample $\boldsymbol{a}_{t_k}$; $\boldsymbol{a}_0^{(t_k)}$ to denote the reconstruction of the signal $\boldsymbol{a}$ obtained from a generic diffusion inverse solver at time t_k; and $\tilde{\boldsymbol{a}}_0^{(t_k)}$ to denote the reconstruction of the signal $\boldsymbol{a}$ obtained from DDIS specifically. We use $\boldsymbol{u}_{\text{obs}}$ systematically to denote the measurements. $\pi \Rightarrow \pi'$ represents convergence in distribution of π to π'. We denote by $L^2(\Omega)$ as the space of square (Lebesgue) integrable functions on domain Ω.

B FUNCTION SPACE DIFFUSION SETUP

Let $\boldsymbol{K}$ be a kernel matrix, with entries $\boldsymbol{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a trace-class, positive definite kernel. For example, $k(\cdot, \cdot)$ can be the radial basis function (RBF) kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2/\ell)$ for a fixed length scale $\ell > 0$. We assume K is positive definite and write $K^{1/2}$ for its symmetric positive definite square root. Let $\sigma_t > \sigma_0 = 0$ for $t > 0$ be a diffusion noise schedule, and assume that $t \mapsto \sigma_t^2$ is continuously differentiable and nondecreasing. We write p_t for the density of $\boldsymbol{a}_t$ so that $\boldsymbol{a}_t | \boldsymbol{a}_0 \sim \mathcal{N}(\boldsymbol{a}_0, \sigma_t^2 \boldsymbol{K})$ and consider the variance exploding (VE) diffusion forward process at $0 < t_0 < \cdots < t_N = T$:

$$\boldsymbol{a}_{t_{k+1}} = \boldsymbol{a}_{t_k} + \sqrt{\sigma_{t_{k+1}}^2 - \sigma_{t_k}^2}\, \boldsymbol{\varepsilon}_k \,, \qquad k = 0, \ldots, N-1 \,, \tag{18}$$

where the $\boldsymbol{\varepsilon}_k \sim \mathcal{N}(0, K)$ are independent across k.

With $g(t) = \sqrt{\frac{d}{dt}\sigma_t^2}$, the continuous-time forward process is given by the VE SDE:

$$d\boldsymbol{a}_t = g(t) K^{1/2} d\boldsymbol{W}_t \,, \qquad 0 \le t \le T \,, \tag{19}$$

where W_t is a n-dimensional Brownian motion. With t decreasing from T to 0, the reverse VE SDE is given by:

$$d\boldsymbol{a}_t = -g(t)^2 \boldsymbol{K} \nabla_{\boldsymbol{a}_t} \log p_t(\boldsymbol{a}_t)\, dt + g(t) \boldsymbol{K}^{1/2}\, d\overline{\boldsymbol{W}}_t \,, \qquad 0 \le t \le T \,, \tag{20}$$

where $\overline{\boldsymbol{W}}_t$ is n-dimensional Brownian motion in reverse time. To obtain the appropriate reverse diffusion process, one learns $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{a}_t, t) \approx \nabla_{\boldsymbol{a}_t} \log p_t(\boldsymbol{a}_t)$ with score matching and then discretizes Eq. (20) at times t_k via, e.g., Euler-Maruyama (Kloeden & Platen, 1992). Additionally, by the function-space analogue of Tweedie's formula (Lim et al., 2025), which after discretization yields the standard multivariate Tweedie's formula (Efron, 2011), we have:

$$D^*(\boldsymbol{a}_t, t) := \mathbb{E}[\boldsymbol{a}_0 | \boldsymbol{a}_t] = \boldsymbol{a}_t + \sigma_t^2 \boldsymbol{K} \nabla_{\boldsymbol{a}_t} \log p_t(\boldsymbol{a}_t) \tag{21}$$

We note that this does not require $\boldsymbol{a}_0$ to have a density, making it compatible with our prior in Sec. 3; under this prior, a density is admitted for all numerical times $t_0 < \cdots < t_N$, yielding that Eq. (21) is well-defined. For a learned score $\boldsymbol{s}_{\boldsymbol{\theta}}$, we denote the denoiser by $D_{\boldsymbol{\theta}}$.

C EXTENDED RELATED WORK

C.1 CLASSICAL INVERSE SOLVERS

The partial differential equation (PDE) inverse problem in Sec. 2 is prevalent across many subfields of science and engineering, including geophysics (Virieux & Operto, 2009), astrophysics (Peng et al., 2024; Christensen & Meyer, 2022), fluid dynamics (Bui-Thanh et al., 2013), imaging (Arridge, 1999), and materials (Avril et al., 2008). Thus, many methods have been developed to solve PDE inverse problems (Belov, 2012). One key approach involves specifying different structured priors, such as

Besov (Dashti et al., 2012) or level set (Iglesias et al., 2016) priors, and sampling from the posterior with a MCMC method such as the preconditioned Crank-Nicolson method (Cotter et al., 2013). While these methods often come with formal guarantees, they can take prohibitively long to sample from the posterior, especially under multimodality or in high dimensions (Guan & Krone, 2007; Ge et al., 2018). Because of the expense of these methods, exact forward models are rarely used in the inner loop, and rather domain-specific emulators are used instead. These forward models are often highly-tuned and problem specific, limiting generalization. For example, the gravitational wave community replaces their expensive numerical relativity simulations with hand-parametrized effective models, and doing so has been a decades-long endeavor (Tiglio & Villanueva, 2022). Recently, multi-fidelity techniques have been leveraged to build emulators in these data-sparse environments(Gorodetsky et al., 2020). Other methods include optimization-based samplers, such as randomize-then-optimize methods (Bardsley et al., 2014), which repeatedly compute the maximum a posteriori estimate under different random perturbations of the observations and prior, as well as ensemble approaches based on the Kalman filter (Iglesias et al., 2013). These methods, however, fail to sample from the correct posterior under nonlinear forward operators and complicated priors (Ernst et al., 2015). Recently a concerted effort has been made in many communities, such as plasma physics (Citrin et al., 2024), to make differentiable forward models that may be placed into optimization loops (Sapienza et al., 2024). We note that many other methods have been developed for the inverse problem in Sec. 2 without taking the Bayesian perspective, such as compressed sensing (Alberti & Santacesaria, 2021), but these only produce point estimates, thus falling outside the scope of our work.

C.2 Physics-Informed Learning

Physics-informed learning has become a widely used approach for accelerating scientific computation (Karniadakis et al., 2021) across diverse fields such as climate modeling (Kashinath et al., 2021) and plasma physics (Qin et al., 2023). Below, we discuss key approaches towards solving forward and inverse problems.

C.2.1 Forward Solvers

Representative approaches to solving the forward problem include physics-informed neural networks (PINNs) (Raissi et al., 2019), which encode the governing PDE into the training objective, and neural operators (Li et al., 2021), which learn mappings between function spaces and can amortize repeated PDE solves. For modeling the forward operator, neural ODEs (Chen et al., 2018) are often leveraged to replace a component of a solver with a neural network. Deterministic formulations of these methods return a single reconstruction rather than samples across the posterior. Probabilistic variants have been developed (Park et al., 2026), but require problem-specific training and additional modeling assumptions. We study methods which can be used across several measurement operators without retraining. Thus, in our setting, we primarily use neural operators as differentiable surrogates for forward PDE operator and as the denoiser for function space diffusion in our experiments. Separately, several approaches have been presented for physics-informed (Bastek et al., 2025) and physics-constrained (Utkarsh et al., 2025) generation; these approaches are complementary but distinct to solving PDE inverse problems with diffusion.

C.2.2 Priors

Diffusion models have seen enormous success in modeling complicated, high-dimensional priors (Rombach et al., 2022). Thus, several recent works have constructed approaches to function space diffusion priors. Kerrigan et al. (2023) develop diffusion models on possibly infinite-dimensional Hilbert spaces using Gaussian measures, and Pidstrigach et al. (2024) similarly formulate diffusion models in infinite dimensions and study well-posedness and approximation bounds. Lim et al. (2025) develop Denoising Diffusion Operators (DDO), where neural operator architectures parameterize the learned score during score matching. These provide a concrete mathematical and algorithmic basis for using diffusion models to model function space priors for use in diffusion-based PDE inverse solvers.

C.2.3 INVERSE SOLVERS

Due to their ability to model complicated priors, diffusion models have been adapted to PDE inverse problems. In our work, we focus on developing diffusion solvers which are agnostic to the choice of measurement operator; that is, one learned prior can be used with many different measurement operators during inference. DiffusionPDE (Huang et al., 2024) replaces the diffusion likelihood $\pi(\boldsymbol{u}_{\text{obs}}|\boldsymbol{a}_t)$ with a Gaussian centered at the denoiser estimate $\hat{\boldsymbol{a}}_0^{(t)}$, and samples from the joint distribution $(\boldsymbol{a}, \boldsymbol{u})$. Notably, DiffusionPDE trains a standard diffusion prior at fixed discretization; instead, FunDPS (Yao et al., 2025) trains a function space prior. More recent work, such as the state-of-the-art DDIS (Lin et al., 2026), has decoupled the physical parameter and solution fields $(\boldsymbol{a}, \boldsymbol{u})$, specifically approximating $\boldsymbol{u}$ with a neural operator surrogate while performing prior learning and posterior sampling on $\boldsymbol{a}$. In our work, we also decouple the learning of $\boldsymbol{a}$ and $\boldsymbol{u}$. DDIS also relaxes the assumptions of DiffusionPDE and FunDPS; instead of approximating the diffusion likelihood with a Gaussian, DDIS approximates $\pi(\boldsymbol{a}_0|\boldsymbol{a}_t)$ with an isotropic Gaussian, as discussed in Sec. 3. Another approach, RED-DiffEq (Shan et al., 2026), performs regularized optimization on the measurement consistency objective, but uses diffusion to encode the regularization; however, this approach is not Bayesian and has not been compared with the strong diffusion posterior samplers discussed above. The concurrent work DiLO (Liu & Lin, 2026) adopts the same decoupling as DDIS, but instead discusses the issue of evaluating the neural operator surrogate on noisy diffusion states $\boldsymbol{a}_t$ instead of clean reconstructions, which the operator was originally trained on. We note that their solution for this issue can easily be incorporated into our input optimization procedure introduced in Sec. 4, and the failure modes of the neural operator surrogate is not the focus of our work. Additionally, the concurrent work FAPS (Shi et al., 2026b) is similar to DDIS, instead using operator flow matching (Shi et al., 2025) to encode the prior and approximating the denoising distribution with an anisotropic Gaussian.

To distinguish these function space inverse solvers, consider two key operations that occur within a reverse sampling step t_k with noisy $\boldsymbol{a}_{t_k}$: (**D**) computing an estimate of the signal $a(\cdot)$ as $\hat{\boldsymbol{a}}_0^{(t_k)} = D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}, t_k)$, where $D_{\boldsymbol{\theta}}$ is the diffusion denoiser and (**M**) enforcing measurement consistency on a reconstruction $\boldsymbol{a}_0^{(t_k)} \approx \hat{\boldsymbol{a}}_0^{(t_k)}$, so that it is consistent with the observations (i.e., $\mathcal{A}(\boldsymbol{a}_0^{(t_k)}) \approx \boldsymbol{u}_{\text{obs}}$). We use the term *output space optimization* to describe methods where (**D**) and (**M**) are *disjoint*, since this amounts to optimizing in the output space of the denoiser, while leaving the denoiser input fixed. From this perspective, DiffusionPDE (Huang et al., 2024), FunDPS (Yao et al., 2025), and Decoupled Diffusion Inverse Solver (DDIS) (Lin et al., 2026) are all output space optimization procedures. We discuss this further in App. E.

In Sec. 3, we demonstrate that output space approaches can fail to update unobserved components of the denoiser estimate appropriately for a PDE inverse problem. We thus use this to motivate *input space optimization* in Sec. 4, where steps (**D**) and (**M**) are *combined*: we iteratively compute the denoiser output (**D**) and optimize the *input* such that measurement consistency is satisfied (**M**). We show that this updates the unobserved components as desired. This results in *backward-consistent* sampling (Alkhouri et al., 2025), where the reconstruction is constrained to be in the denoiser's range. We note that our approach is different, motivated the setting of PDE inverse problems where priors can be sharply multimodal and with a new approach to allocate input optimization steps motivated by our theory. Additionally, the concurrent work FAPS (Shi et al., 2026b) uses its anisotropic Gaussian covariance to precondition the measurement gradient. We note that input space optimization can be viewed as *preconditioned* output space optimization, as discussed in App. J. Note that our preconditioner differs from FAPS, and furthermore is grounded in the theory of PDE inverse problems and does not require additional estimation from training data after the diffusion model has been trained.

The computer vision literature also contains several approaches to address the unobserved component of the reconstruction while solving the inverse problem. Most notably, DDNM (Wang et al., 2023) assumes a linear forward operator and does diffusion only in nullspace; NullFlow (Shi et al., 2026a) extends this approach to one-step with flow matching. Note that these approaches do not solve the issue that DDIS suffers from discussed in Prop. 1. In particular, the nullspace component of the reconstruction is still the same as that of the denoiser. Separately, Posterior-DDIM (Jiao et al., 2026) performs ODE-based diffusion sampling in each singular direction separately, incorporating measurement consistency in a singular direction only when the diffusion signal to noise ratio is small. Importantly, this approach assumes a linear forward operator so that the singular value decomposition

is easily computable. However, PDE inverse problems can have highly nonlinear forward operators, e.g., Darcy flow as considered in our experiments in Sec. 5. One can obtain a local approximation with the Jacobian, but this still comes at a high computational expense and may not have the same desirable properties.

Finally, note that there have been several recent approaches developed to use diffusion for data assimilation (Andrae et al., 2026; Jia et al., 2026). While data assimilation is closely related to Bayesian inverse problems, it considers a sequential setting; rather than recovering a fixed coefficient field $\boldsymbol{a}$ from measurements $\boldsymbol{u}_{\text{obs}}$, data assimilation repeatedly propagates a state $\boldsymbol{a}_0^{(\ell)}$ and assimilates new observations $\boldsymbol{u}_{\text{obs}}^{(\ell)}$. Accordingly, recent generative data assimilation work focuses on combining learned generative priors with state transition operators. These approaches are complementary to our work. Furthermore, moment-matching posterior samplers like DPS (Chung et al., 2023) or MMPS (Rozet et al., 2024) have been adopted for use in data assimilation (Chen et al., 2025; Savary et al., 2026); backward-consistent sampling can be adopted as well, and we leave a comprehensive study of this to future work.

C.3 Theoretical Guarantees

Classical inverse solver theory provides guarantees for recovering an unknown signal from incomplete observations. For deterministic inverse problems, where one recovers a point estimate, regularization theory extensively studies the existence, stability, and convergence of solutions, especially with linear forward operators (Engl et al., 1996). Bayesian inverse solver theory studies when the posterior is well-defined and when the problem is well-posed and stable (Stuart, 2010); theory for MCMC methods is likewise used to study MCMC-based solvers for these problems (Hairer et al., 2014), including using diffusion priors (Wu et al., 2024b). Due to the prevalence of these diffusion priors in recent inverse solvers, recent works likewise have developed theory for diffusion inverse solvers. For Gaussian priors, approximating the denoising distribution exactly by first- and second-moment Tweedie's formula can recover the posterior (Boys et al., 2024). Other recent work obtains insights into the bias and stability of diffusion posterior samplers (Delgadino et al., 2026); the robustness of these samplers to likelihood misspecification and outliers (Yang et al., 2026); connections to projected gradient descent and its convergence theory (Leong & Traonmilin, 2026); convergence to the posterior (Jiao et al., 2026); and investigating how this relates to maximum a posteriori estimation (Xu et al., 2025). Notably, (Burns & Fridovich-Keil, 2026) study multimodal priors; however, they focus on a different failure mode of diffusion posterior samplers, specifically issues with likelihood approximations such as those made by DPS. Additionally, several works have used multimodal Gaussian mixture priors to empirically study the behaviors of diffusion posterior samplers (Crafts & Villa, 2025; Yismaw et al., 2025); however, these works do not study the PDE setting or the unobserved components of the denoiser. Much of the literature studies the finite-dimensional image inverse problem setting; for the function space setting present in our PDE inverse problems, FunDPS (Yao et al., 2025) develops an analogue of Tweedie's formula while DDIS (Lin et al., 2026) obtains theoretical results characterizing why learning the coupled prior $(\boldsymbol{a}, \boldsymbol{u})$ is problematic, motivating the learning of the prior over only $\boldsymbol{a}$ done in both their and our algorithms. More broadly, there has been significant work on how diffusion models select modes under multimodal priors (Wu et al., 2024a; Ashiq et al., 2026). Our work adds to this literature in the PDE inverse problem setting; we study when and why diffusion solvers based on output space optimization fail in the PDE setting, where multimodal priors naturally arise, and instantiate a solution based on input space optimization, validating our results empirically. We also study what unique conditions on the measurements and nullspace components of different prior modes arise due to the PDE setting in Prop. 8 in App. K.11 and connect our results to preconditioning in App. J.

The effectiveness of conditioning in diffusion models depends on when and how guidance is applied throughout the reverse process; while classifier and classifier-free guidance are frequently implemented using a constant guidance scale, several works show that this is suboptimal across diffusion noise levels (Papalampidi et al., 2026). In the setting of Bayesian inverse solvers based on diffusion models, there has been work done on designing schedules for likelihood or measurement guidance across the diffusion sampling process. For example, Jacobian-aware posterior sampling (Hen et al., 2026) adapts the step size of the likelihood gradient based on the denoiser Jadcobian, while the recent work TriPS (Bang et al., 2026) adjusts this step size for conditional priors based on the classifier-free guidance schedule and added stochasticity in the sampling process. Additionally, Delgadino et al.

(2026) show that DPS guidance can become numerically unstable towards the end of the reverse process, providing a principled explanation for early stopping the DPS measurement-consistency guidance in the sampling process. We add to this body of work, designing a principled approach to scheduling input optimization steps across noise levels based on our theoretical analysis of an inverse problem on Poisson's equation. Notably, we control a different quantity than, e.g., gradient step size or guidance weight; instead, we control the *computational effort* used to enforce measurement consistency at each diffusion noise level, done so by a function that automatically allocates more steps on critical noise levels during inference.

D Limitations & Future Directions

Limitations include that FunBCS is reliant on the accuracy of the neural operator surrogate $\mathcal{G}_{h,\phi}$ and the diffusion prior $\pi(\boldsymbol{a})$. Additionally, our current input optimization scheduler requires task-specific calibration. Future directions include addressing these limitations, improving our method further for discontinuous inverse problems by, e.g., using discontinuous bases (Iglesias et al., 2016), and extending input space optimization to PDE data assimilation (Chen et al., 2025).

E Overview of Recent Diffusion PDE Inverse Solvers

We provide an overview of the recurrences of diffusion PDE inverse solvers, to make clear how existing methods perform output space optimization (with separate (**D**) and (**M**) steps) while our method performs input optimization (combined (**D**) and (**M**) steps). We present these all for fixed diffusion times $t_k > t_{k-1}$.

DiffusionPDE/FunDPS recurrence. Both DiffusionPDE (Huang et al., 2024) and FunDPS (Yao et al., 2025) extend diffusion posterior sampling (Chung et al., 2023) to function spaces, which uses the following recurrence:

$$\boldsymbol{a}_{t_k-1}^{(P)} = \tilde{\boldsymbol{a}}_{t_k-1}^{(P)} - \zeta_k \nabla_{\boldsymbol{a}_{t_k}^{(P)}} \|\mathcal{A}_h \hat{\boldsymbol{a}}_0^{(P,t_k)} - \boldsymbol{u}_{\text{obs}}\|_2^2 \,, \tag{22}$$

where:

$$\tilde{\boldsymbol{a}}_{t_k-1}^{(P)} = \boldsymbol{a}_{t_k}^{(P)} + \Delta_{t_k} \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(P)}, t_k) + \sqrt{\Delta_{t_k}} \boldsymbol{\varepsilon}_k^{(P)} \,. \tag{23}$$

and $\Delta_{t_k} = t_k - t_{k-1}$ with $\hat{\boldsymbol{a}}_0^{(P,t_k)} := D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(P)}, t_k)$ with $\boldsymbol{\varepsilon}_k^{(P)}$ independent.

Remark 8. At a timestep t_k, one first computes $\hat{\boldsymbol{a}}_0^{(P,t_k)}$ (i.e., **(D)**); this can then be used in the measurement consistency term in Eq. (22) as well as to estimate $\boldsymbol{s}_{\boldsymbol{\theta}}$ by Tweedie's formula per Eq. (4), used in Eq. (23). DPS then performs a renoising-type step in Eq. (23), and subsequently enforces measurement consistency (i.e., performs (**M**) in Eq. (22). We note that these two steps are separate at diffusion timestep t_k, so that measurement consistency is enforced by optimizing over the ouput space of the denoiser. That is, methods such as DiffusionPDE and FunDPS perform output space optimization.

DDIS recurrence. As presented in Eq. (6), the DDIS recurrence is given by:

$$\boldsymbol{a}_{t_{k-1}}^{(D)} = \tilde{\boldsymbol{a}}_0^{(t_k)} + \sigma_{t_{k-1}} \boldsymbol{\varepsilon}_k^{(D)}, \qquad \tilde{\boldsymbol{a}}_0^{(t_k)} \sim q_{D,t_k}(\cdot | \boldsymbol{a}_{t_k}^{(D)}, \boldsymbol{u}_{\text{obs}}),\ \boldsymbol{\varepsilon}_k^{(D)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K}) \,, \tag{24}$$

with

$$q_{\text{D},t_k}(\boldsymbol{a}_0 \mid \boldsymbol{a}_{t_k}^{(D)}, \boldsymbol{u}_{\text{obs}}) \propto \exp\left\{ -\frac{\|\mathcal{A}_h \boldsymbol{a}_0 - \boldsymbol{u}_{\text{obs}}\|^2}{2\gamma^2} - \frac{\|\boldsymbol{a}_0 - \hat{\boldsymbol{a}}_0^{(D,t_k)}\|^2}{2\sigma_{t_k}^2} \right\} \,. \tag{25}$$

where $\hat{\boldsymbol{a}}_0^{(D,t_k)} := D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)}, t_k)$ and $\varepsilon_k^{(D)}$ are drawn independently.

Remark 9. At a timestep t_k, one first computes $\hat{\boldsymbol{a}}_0^{(D,t_k)}$ (i.e., **(D)**). DDIS then incorporate this into Eq. (25) to sample from q_{D,t_k}; from the perspective of Langevin dynamics, this is precisely optimizing in the output space of the denoiser by enforcing measurement consistency (**M**) with the first term and ensuring the reconstruction does not deviate too far from the denoiser estimate per the

second term. Subsequently, DDIS renoises up to timestep t_{k-1} per Eq. (24). Computing the denoiser output and enforcing measurement consistency are clearly separated. Thus, DDIS performs output space optimization.

FunBCS recurrence. As presented in Eq. (13), the FunBCS recurrence is given by:

$$\boldsymbol{a}_{t_{k-1}}^{(B)} = D_{\boldsymbol{\theta}}(\boldsymbol{v}(s), t_k) + \sigma_{t_{k-1}} \boldsymbol{\varepsilon}_k^{(B)} , \qquad \boldsymbol{\varepsilon}_k^{(B)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{K}) , \tag{26}$$

where $\boldsymbol{v}(s)$ is a solution to the gradient flow:

$$\frac{d\boldsymbol{v}(s)}{ds} = -\nabla_{\boldsymbol{v}} \frac{\|\mathcal{O}_h \circ \mathcal{G}_{h,\phi} \circ D_{\boldsymbol{\theta}}(\boldsymbol{v}, t_k) - \boldsymbol{u}_{\text{obs}}\|^2}{2\gamma^2}\bigg|_{\boldsymbol{v}=\boldsymbol{v}(s)} , \qquad \boldsymbol{v}(0) = a_{t_k}^{(B)}. \tag{27}$$

and $\boldsymbol{\varepsilon}_k^{(B)}$ are drawn independently.

Remark 10. At a fixed diffusion timestep t_k, per Eq. (27), for an optimization step s_i, we alternate between computing the denoiser output $D_{\boldsymbol{\theta}}(\boldsymbol{v}(s_i, t_k)$ and finding the input $\boldsymbol{v}(s_i + \Delta_s)$ so that measurement consistency is satisfied. This thus optimizes in the *input* space of the denoiser and couples steps (**D**) and (**M**). Hence, our method FunBCS is an input space optimization procedure.

F Experimental Details

Tasks. We use the four PDE inverse problems of DDIS (Lin et al., 2026), whose data were generated by DiffusionPDE (Huang et al., 2024) with the solvers of FNO (Li et al., 2021); we use the normalized version provided with DDIS's public codebase. Every problem is posed on the unit square $\Omega = (0, 1)^2$, discretized on a 128×128 grid. In each problem, we recover the coefficient field a from 500 noiseless observations of the solution u at random grid points, about 3% of the grid. The observation locations are drawn anew for each test sample and are the same for FunBCS, DDIS and FunDPS, whereas DiffusionPDE's public code uses one fixed set of locations for every sample. Below, ∇^2 denotes the Laplacian, and a Gaussian random field with parameters (α, τ) has covariance proportional to $(-\nabla^2 + \tau^2 I)^{-\alpha}$. The coefficient fields of Poisson, Helmholtz and Darcy flow are all built from such a field with $(\alpha, \tau) = (2, 3)$ and zero Neumann boundary conditions.

Poisson. The source term a is the Gaussian random field above, and

$$\nabla^2 u(\boldsymbol{x}) = a(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega, \qquad u(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \partial\Omega.$$

Helmholtz. The source term a is drawn as for Poisson and set to zero on $\partial\Omega$, and with wavenumber $k = 1$,

$$\nabla^2 u(\boldsymbol{x}) + k^2 u(\boldsymbol{x}) = a(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega, \qquad u(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \partial\Omega.$$

Navier–Stokes. The vorticity $w = \nabla \times \boldsymbol{v}$ of an incompressible velocity field $\boldsymbol{v}$ on the periodic domain Ω evolves as

$$\partial_t w + \boldsymbol{v} \cdot \nabla w = \nu \nabla^2 w + f, \qquad \nabla \cdot \boldsymbol{v} = 0, \qquad \boldsymbol{x} \in \Omega,\ t \in (0, T],$$

with viscosity $\nu = 10^{-3}$ (Reynolds number 1,000) and forcing $f(\boldsymbol{x}) = 0.1\big(\sin(2\pi(x_1 + x_2)) + \cos(2\pi(x_1 + x_2))\big)$. The coefficient field is the initial vorticity $a = w(\cdot, 0)$, a periodic Gaussian random field with $(\alpha, \tau) = (2.5, 7)$, and the solution is the vorticity $u = w(\cdot, T)$ at $T = 1$.

Darcy flow. The permeability a equals 12 where a Gaussian random field drawn as for Poisson is nonnegative and 3 elsewhere, so it is piecewise constant, and

$$-\nabla \cdot \big(a(\boldsymbol{x}) \nabla u(\boldsymbol{x})\big) = 1, \quad \boldsymbol{x} \in \Omega, \qquad u(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \partial\Omega.$$

DDIS's public codebase supports Darcy flow, but DDIS does not report it, so we include it because its piecewise-constant coefficient makes the prior multimodal.

Evaluation. All methods are evaluated on the same 100 test samples. On Poisson, Helmholtz and Navier-Stokes, we report the relative ℓ_2 error $\|\hat{a} - a\|_2/\|a\|_2$ in percent. On Darcy flow, we threshold the reconstruction at 7.5, the midpoint of the two coefficient values, and report the percentage of misclassified pixels (Huang et al., 2024). The spectral energy error, from Lin et al. (2026), E_s compares the radially averaged power spectra $\hat{P}(k)$ and $P(k)$ of the reconstruction and of the ground truth. We take the geometric mean of $|\hat{P}(k) - P(k)|/P(k)$ over wavenumbers k, and then average over the test samples.

Models and samplers. FunBCS uses the same pretrained diffusion prior, neural operator surrogate and noise schedule as DDIS Lin et al. (2026). DDIS uses the settings of DDIS's public codebase: 100 noise levels, 20 Langevin steps per level, and the likelihood weights recommended in the paper. for Darcy flow, which DDIS does not report, we tuned and chose the weight 30 from $\{5, 10, 15, 20, 30\}$. FunBCS uses $N = 10$ noise levels and optimizes the denoiser input with Adam at learning rate $\gamma = 0.03$. Its scheduling function (App. G) uses the selection windows $(\sigma_{\text{low}}, \sigma_{\text{high}}) = (0, 0.711)$ on Poisson, $(0, 0.335)$ on Helmholtz, $(0, 0.144)$ on Navier–Stokes and $(0.335, 2.6)$ on Darcy flow. FunBCS-AC uses $M = NK = 800$ optimization steps per sample, and FunBCS-EF uses $M^\star = 607$, 506, 406 and 488 steps on the four tasks, respectively. We note that these $M^\star$ budgets are not hardcoded, but instead dynamically determined by our method during inference.

Baselines. DiffusionPDE (Huang et al., 2024) trains a diffusion model on the joint distribution of the coefficient and solution fields at a fixed resolution, and guides its sampling with the observation misfit and the PDE residual, evaluated on the denoiser estimate. FunDPS (Yao et al., 2025) follows the same approach with a joint diffusion prior in function space built from neural operators. DDIS (Lin et al., 2026) learns a diffusion prior on the coefficient field only, predicts the solution with a neural operator surrogate, and enforces the observations with annealed Langevin sampling in the output space of the denoiser. We run each baseline with its authors' public code and default settings. For the scaling curves of Figs. 3 and 6, we run every method at $0.5\times$, $1\times$, $2\times$ and $4\times$ its default budget and change nothing else: FunBCS with 400, 800, 1,600 and 3,200 optimization steps, distributed over its $N = 10$ noise levels by the scheduling function; DDIS with 50, 100, 200 and 400 noise levels of 20 Langevin steps each, that is, 1,000 to 8,000 Langevin steps; and FunDPS with 250, 500, 1,000 and 2,000 sampling steps. DiffusionPDE, which is much slower, is run with 500, 2,000 and 3,000 sampling steps, where 2,000 is its default.

Choosing N and K. We fix the max budget of FunBCS at $NK = 800$ optimization steps. In order to determine the optimal N, assuming a uniform allocation of K inner steps per outer level, we had to choose how to split between the number of noise levels N and the number of steps K per level. Figure 5 compares three splits on Poisson and Darcy flow. These runs place the same K steps on every noise level and do not use the scheduling function at all, so that the choice of N and K is not confounded with how the steps are allocated. The split $N = 10$, $K = 80$ reaches the lowest error on both tasks, and we use it for every task. The scheduling function then redistributes these 800 steps across the $N = 10$ noise levels (App. G).

Timing. We measure seconds per sample on a single NVIDIA A100 GPU with one process per GPU, at batch size 25 for FunBCS, DDIS and FunDPS and at batch size 1 for DiffusionPDE. DiffusionPDE's public code reconstructs one test sample at a time, with a single ground truth, observation mask and guidance loss per run, so we time it unmodified at batch size 1. The time covers the whole sampling process, including all denoiser and surrogate evaluations. Each optimization step of FunBCS differentiates through both the denoiser and the surrogate, so it costs more than a Langevin step of DDIS, which differentiates through the surrogate only. FunBCS, however, uses 10 noise levels instead of 100, which DDIS uses.

Scaling with compute. Figure 6 shows the error of every method against its time per sample on Helmholtz, Navier–Stokes and Darcy, as Fig. 3 does for Poisson. As on Poisson, FunBCS is more accurate than every baseline at every budget, and at about 8 seconds per sample it is more accurate than every baseline at any of the budgets we tested.

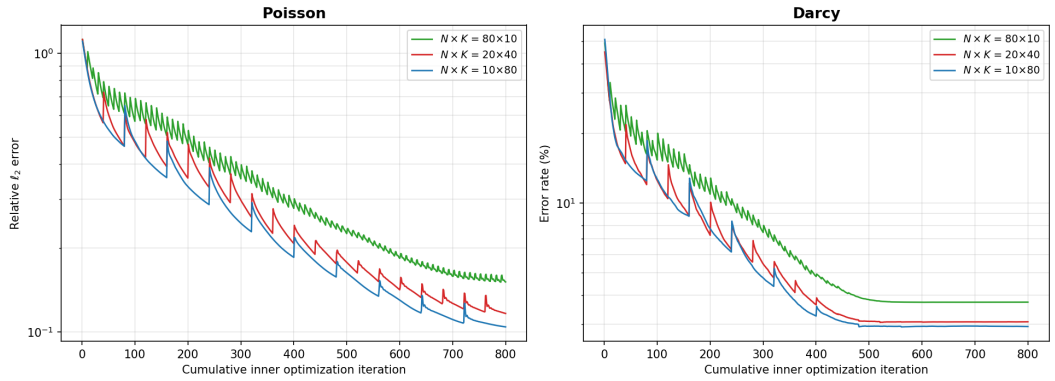


Figure 5: Reconstruction error of FunBCS during sampling for three splits $N \times K$ of 800 optimization steps, with the same K at every noise level and without the scheduling function, on 50 samples: relative ℓ_2 error on Poisson (left) and pixel error rate on Darcy flow (right). The error jumps up at the end of each noise level, when the sample is re-noised.

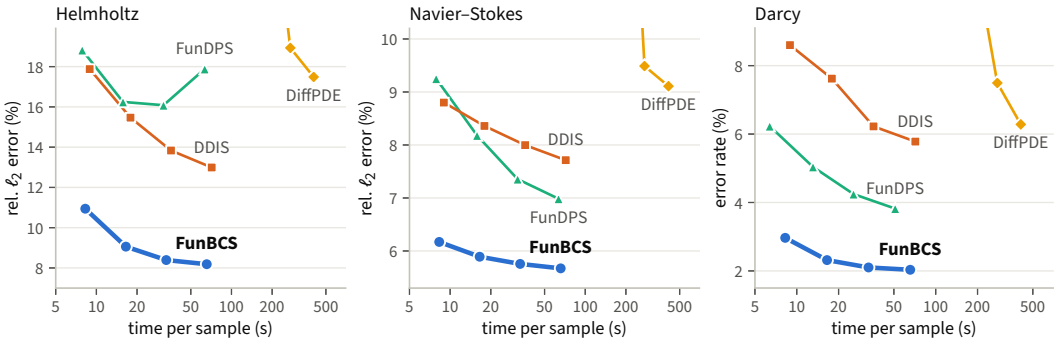


Figure 6: Scaling behavior of all methods with increasing inference time per sample on Helmholtz, Navier–Stokes and Darcy, as in Fig. 3 for Poisson. Darcy reports the pixel error rate. All methods are run using the same hardware.

G The Proposed Scheduling Function

A uniform schedule gives every noise level the same number of optimization steps, although the analysis of Sec. 4.2 shows that steps matter unequally across noise levels: every level needs a few, so that a lean towards the wrong mode is corrected before it is carried to lower noise levels (Prop. 4), and the remaining steps are worth the most inside the selection window (Prop. 5). The scheduling function therefore treats the optimization steps of algorithm 1 as a total budget M and allocates it across noise levels, keeping a floor at every level and concentrating the remainder in the selection window. Figure 7 shows the resulting allocations, and that withholding the steps of a single noise level hurts most where the scheduler concentrates them. We define the allocation below, then ablate it (App. G.1), describe how the window is tuned (App. G.2), and apply it to DDIS (App. G.3).

Effort weights. The scheduling function assigns each noise level σ an effort weight through a smooth window whose only task-specific parameters are the two edges $\sigma_{\text{low}} \leq \sigma_{\text{high}}$ of the selection window:

$$\omega(\sigma) = \Big[1 + (\sigma/\sigma_{\text{high}})^{\beta}\Big]^{-1} \Big[1 + (\sigma_{\text{low}}/\sigma)^{\beta}\Big]^{-1}. \tag{28}$$

The weight is close to one inside the window and decays polynomially outside it, at a rate set by the exponent $\beta > 0$. Setting $\sigma_{\text{low}} = 0$ removes the lower edge, setting $\sigma_{\text{high}} = \infty$ removes the upper edge, and $\beta \to \infty$ recovers a hard window. Because ω depends on the noise level rather than on the index k, the same edges apply to any number and spacing of noise levels.

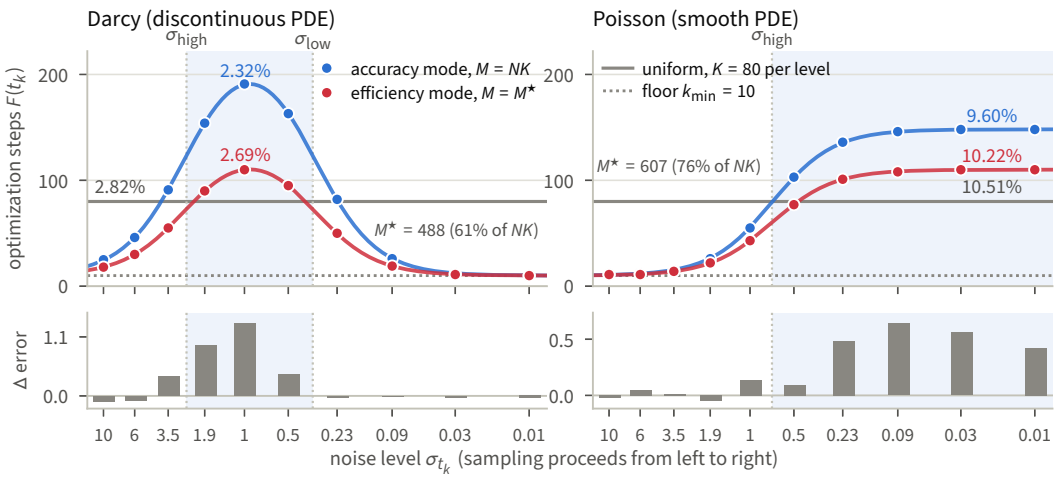


Figure 7: **Top**: Optimization steps per noise level under the scheduling function (App. G). The accuracy mode (blue) reallocates the $NK = 800$ steps of the uniform allocation (grey) towards the selection window (shaded), and the efficiency mode (red) cuts the budget to $M^\star$; every noise level keeps at least $k_{\min} = 10$ steps (dotted). Numbers give the resulting error (Table 1). **Bottom**: increase in error, in percentage points, when a single noise level receives no optimization steps, which is largest where the scheduler places its steps.

Allocation. Given a total budget of M optimization steps, the scheduler reserves a floor of $k_{\min}$ steps at every noise level and distributes the remainder in proportion to the effort weights $\omega_k := \omega(\sigma_{t_k})$:

$$F(t_k) = k_{\min} + \left(M - Nk_{\min}\right) \frac{\omega_k}{\sum_{j=1}^{N} \omega_j}, \qquad k = 1, \ldots, N. \tag{29}$$

We round each share down and give the remaining steps, one each, to the noise levels with the largest remainders, so that $\sum_k F(t_k) = M$ exactly. A uniform allocation of K steps per noise level corresponds to $\omega \equiv 1$, and at the same budget $M = NK$, Eq. (29) moves steps towards the selection window without changing the cost of the sampler.

Accuracy and efficiency modes. Since the effort weights identify the noise levels on which optimization steps matter, the scheduler also determines a natural budget below NK:

$$M^\star = Nk_{\min} + (K - k_{\min}) \frac{\sum_j \omega_j}{\tau \max_j \omega_j}, \qquad \tau \in (0, 1], \tag{30}$$

capped at NK. At this budget, Eq. (29) assigns exactly K steps to a noise level whose weight is a fraction τ of the peak, so every noise level with at least this weight keeps at least as many steps as under the uniform allocation, and the saving comes from the noise levels outside the window. We call $M = NK$ the *accuracy mode* and $M = M^\star$ the *efficiency mode*, and write FunBCS-AC and FunBCS-EF for FunBCS in these two modes. For every run, FunBCS-AC seeks to non-uniformly disperse the M total optimization steps in order to reach higher accuracy without changing the budget, while FunBCS-EF seeks to cut as much budget as possible while maintaining the similar accuracy as uniform allocation. During inference, one can easily interpolate between the two modes with no tuning as the two modes differ only in the budget.

Shared constants. All FunBCS experiments use $\beta = 2$, $k_{\min} = 10$ and $\tau = 0.7$ on every PDE and in both modes, with $N = 10$ noise levels and $K = 80$, so that the accuracy mode has a budget of 800 steps. Only the window edges are chosen per PDE (App. G.2): $(\sigma_{\text{low}}, \sigma_{\text{high}}) = (0, 0.711)$ on Poisson, $(0, 0.335)$ on Helmholtz, $(0, 0.144)$ on Navier–Stokes and $(0.335, 2.6)$ on Darcy.

G.1 Scheduler Ablations

Table 1 compares FunBCS with the scheduler (FunBCS-AC and FunBCS-EF) against FunBCS without it, which places $K = 80$ optimization steps on each of its $N = 10$ noise levels. At the same

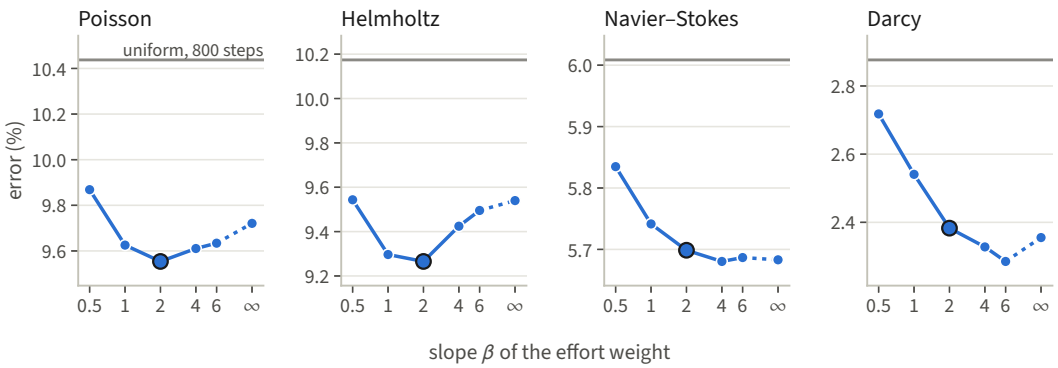


Figure 8: Error of FunBCS in the accuracy mode on 50 samples as the slope β of Eq. (28) varies, at the selected window edges; ∞ denotes a hard window, and the outlined marker is the value used, $\beta = 2$. The grey line is the uniform allocation of the same 800 steps. Darcy reports the pixel error rate and the other tasks the relative ℓ_2 error.

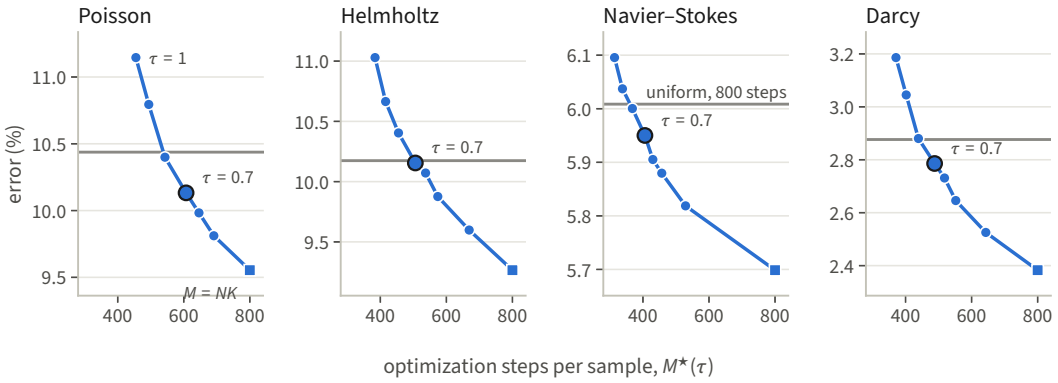


Figure 9: Error of FunBCS on 50 samples against the budget $M^\star$ resolved by Eq. (30) for $\tau = 1, 0.9, 0.8, 0.7, 0.65, 0.6$ and 0.5, from left to right. The outlined marker is the value used, $\tau = 0.7$, the square is the accuracy mode ($M = NK = 800$), and the grey line is the uniform allocation at 800 steps. On Poisson, $\tau = 0.5$ reaches the cap and coincides with the accuracy mode.

budget of 800 steps, the accuracy mode lowers the error by 10% on average, and on every task, at unchanged cost. The efficiency mode removes 37% of the steps on average, which makes FunBCS 1.6× faster, with 2% lower error on average. On Poisson, Helmholtz, Navier–Stokes and Darcy, it uses 607, 506, 406 and 488 of the 800 steps, which take 12.6, 10.5, 8.4 and 10.1 seconds per sample; Table 1 reports their mean. Even without the scheduler, FunBCS is more accurate than every baseline on every task, and faster than DDIS.

Varying the shared constants. Figures 8 to 10 vary the three constants that are shared across tasks, on 50 samples with everything else fixed. The slope β controls how sharply the effort weight of Eq. (28) falls off at the edges of the window. As $\beta \to 0$, the weight becomes flat and the allocation approaches the uniform one; as $\beta \to \infty$, the window becomes a hard box, and every noise level outside it keeps only the floor. We use $\beta = 2$, which gives the lowest error averaged over the four tasks. The constant τ controls the budget of the efficiency mode through Eq. (30). At $\tau = 1$, the noise level with the largest weight keeps the K steps of the uniform allocation and all other levels receive fewer; lowering τ scales the steps above the floor by $1/\tau$, until $M^\star$ reaches the budget NK of the accuracy mode. The floor $k_{\min}$ is the number of steps that every noise level keeps regardless of its weight. Without a floor, the noise levels outside the window receive almost no steps, while a large floor moves the allocation back towards the uniform one.

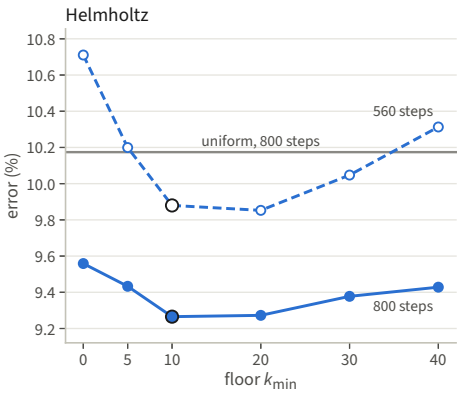


Figure 10: Error of FunBCS on Helmholtz on 50 samples as the floor $k_{\min}$ varies, at 800 and at 560 steps. Outlined markers are the value used, $k_{\min} = 10$, and the grey line is the uniform allocation at 800 steps.

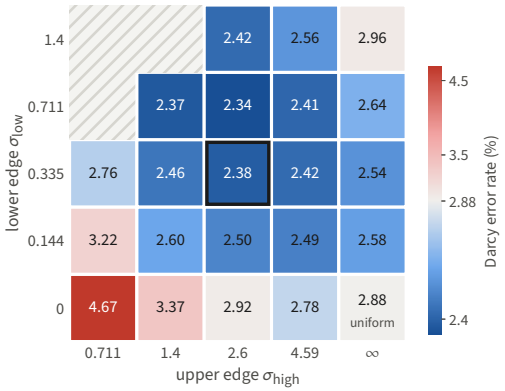


Figure 11: Darcy error rate (%) of FunBCS on 50 samples for each window; $\sigma_{\text{low}} = 0$ and $\sigma_{\text{high}} = \infty$ remove an edge, so the bottom-right cell is the uniform allocation. Blue cells beat it, and the outlined cell is the selected window.

Algorithm 2 Finding the selection window for each PDE

Require: A few samples of the PDE (outside the training and testing sets); set of candidate window bounds $\mathcal{S}$, the midpoints $\sqrt{\sigma_{t_k}\sigma_{t_{k+1}}}$ between consecutive noise levels; a tolerance ϵ for noise in the measured error.

1: Run every configuration below with FunBCS (algorithm 1) in the accuracy mode, using the shared β and $k_{\min}$.
2: Measure the error e_{uni} of the uniform allocation.
3: **Upper bound.** For each $s \in \mathcal{S}$, measure the error $e(s)$ of the window $(0, s)$.
4: Set σ_{high} to the s with the lowest error, or to ∞ (no upper bound) if no window beats the uniform allocation. If σ_{high} is the smallest or largest candidate, also try the next midpoint beyond it and keep the better of the two.
5: **Lower bound.** For each $s \in \mathcal{S}$ below σ_{high}, measure the error $e'(s)$ of the window $(s, \sigma_{\text{high}})$, and let $s^\star$ be the s with the lowest error.
6: Set $\sigma_{\text{low}} = s^\star$ if $e'(s^\star)$ is lower than the error of the window $(0, \sigma_{\text{high}})$ by more than ϵ, and $\sigma_{\text{low}} = 0$ (no lower bound) otherwise.
7: **return** $(\sigma_{\text{low}}, \sigma_{\text{high}})$

G.2 Tuning the Selection Window

Since β, $k_{\min}$ and τ are shared across tasks, only the two edges of the selection window are set for a new PDE. We choose them with the grid search of algorithm 2, run with FunBCS in the accuracy mode. The candidate edges are the geometric midpoints between consecutive noise levels, roughly a factor of two apart; the upper edge is selected first, and a lower edge is kept only if it lowers the error further. Each candidate window is evaluated on 20 samples, so the search, which needs at most nine runs, takes about 50 minutes per PDE on a single A100 GPU.

Sensitivity to the window edges. A coarse grid suffices because the error depends on the shape of the window far more than on the exact position of its edges. Figure 11 shows this on Darcy, the only task whose selected window has both edges. Moving one or both edges of the selected window by one grid step, a factor of about two in σ, keeps the error rate between 2.34 and 2.60%, against 2.88% for the uniform allocation, and 12 of the 13 windows with both edges improve on the uniform allocation. A window of the wrong shape, in contrast, does not: three of the four one-sided low-noise windows that suit the smooth tasks are worse than uniform on Darcy, with error rates up to 4.67%. The selection window is therefore not a sensitive choice. Once its shape is right, each edge only needs to be located to within a grid step, which the coarse search of algorithm 2 does in a few short runs.

G.3 Generalizing Scheduling Function to DDIS

We evaluate whether the scheduling function generalizes to DDIS, an output space optimization method. It significantly improves the speeed and also the accuracy of DDIS after only a short tuning of the window $[\sigma_{\text{low}}, \sigma_{\text{high}}]$. DDIS spends its budget on the Langevin steps with which it samples Eq. (5) at each of its $N = 100$ noise levels. We allocate these steps with Eq. (29), using $K = 20$ steps per noise level for the uniform budget of $2{,}000$ steps, a floor of $k_{\min} = 2$, and the same β and τ as for FunBCS. The floor differs from that of FunBCS only because DDIS takes fewer steps per noise level: we keep the same fraction, one eighth of the steps of each noise level (10 of $K = 80$ for FunBCS), which gives $k_{\min} = 2$ for $K = 20$ after rounding down. The edges are chosen with the same search (algorithm 2), which takes under an hour per PDE on a single A100 GPU. On Poisson, Helmholtz and Navier-Stokes it returns a one-sided low-noise window, $(0, 0.335)$ on Poisson and Helmholtz and $(0, 0.144)$ on Navier–Stokes, identical to or one grid step from the window of FunBCS. On Darcy it returns the window $(0.144, \infty)$.

Table 2 reports the results on the 100 test samples of Table 1. In the accuracy mode, the scheduler lowers the error of DDIS by 8% on average at unchanged cost, and on all four tasks. In the efficiency mode, it makes DDIS $1.3\times$ faster on average while lowering its error by 1% on average. On Darcy, the window covers most noise levels of DDIS, so $M^\star$ reaches the full budget and the two modes coincide. The time falls by less than the number of steps because the probability-flow solve that DDIS runs at every noise level does not depend on the budget.

Table 2: DDIS with and without the scheduling function on the 100 test samples of Table 1. The uniform allocation is DDIS as in Table 1, with $K = 20$ Langevin steps on each of its $N = 100$ noise levels. The accuracy mode redistributes the same $2{,}000$ steps, and the efficiency mode cuts steps while maintaining baseline accuracy. On Darcy, $M^\star$ reaches the full budget, so the two modes coincide. Errors are mean $\pm$ standard deviation over the test samples.

	Poisson		Helmholtz		Navier–Stokes		Darcy	
DDIS	rel. ℓ_2	s/sample	rel. ℓ_2	s/sample	rel. ℓ_2	s/sample	err. rate	s/sample
uniform allocation	15.76±5.08	18.0	15.47±4.34	18.0	8.36±1.55	18.0	7.48±3.89	18.0
accuracy mode	**14.68**±4.71	18.0	**13.95**±4.08	18.0	**8.08**±1.48	18.0	**6.64**±3.62	18.0
efficiency mode	15.56±5.00	**13.1**	16.66±6.10	**13.1**	8.28±1.56	**11.3**	**6.64**±3.62	18.0

H Navier–Stokes Spectral Analysis

Navier–Stokes is the only task where our two metrics disagree: our method improves relative ℓ_2 over DDIS but worsens the spectral energy error E_s (Table 1). This section examines what drives that gap.

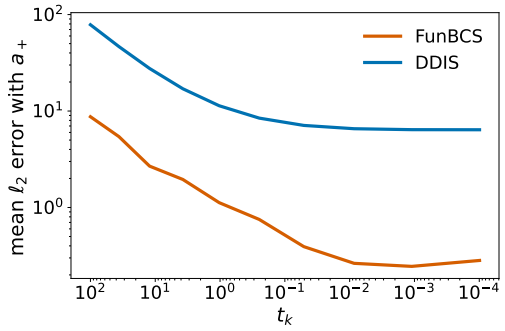
The two metrics can disagree because E_s compares radially averaged power spectra, and is therefore phase-blind. It measures whether a reconstruction carries the right amount of energy at each spatial frequency, but not whether that energy sits in the right places. A reconstruction whose high-frequency content is entirely unrelated to the ground truth can still score well on E_s, as long as its spectrum has roughly the right shape.

To tell these two cases apart, we take the 2D Fourier transform of the reconstruction and of the ground truth, group the modes into radial bins by wavenumber k, and compute a normalized inner product between the two sets of coefficients in each bin,

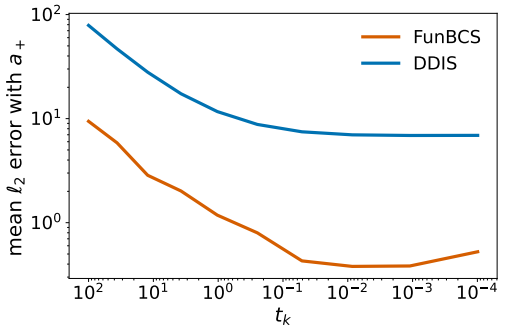
$$\mathcal{K}(k) \;=\; \frac{\left|\sum_{\mathbf{k}\in S_k} \overline{\hat{A}(\mathbf{k})}\, A(\mathbf{k})\right|}{\sqrt{\sum_{\mathbf{k}\in S_k} |\hat{A}(\mathbf{k})|^2}\,\sqrt{\sum_{\mathbf{k}\in S_k} |A(\mathbf{k})|^2}}, \tag{31}$$

where $\hat{A}$ and A are the Fourier coefficients of the reconstruction and the ground truth, $\overline{(\cdot)}$ denotes complex conjugation, and S_k collects every mode falling in radial bin k, across all test instances.

This is a correlation coefficient computed in the Fourier domain, and Cauchy–Schwarz places it in $[0, 1]$. It equals 1 when the reconstructed modes match the true modes in both amplitude and phase, and falls toward 0 when the two are unrelated, no matter how much energy the reconstruction places at that frequency.



(a) Nullspace error to $\boldsymbol{a}_+$. FunBCS (orange) performs significantly better than DDIS (blue).



(b) Cumulative mean error to $\boldsymbol{a}_+$. FunBCS (orange) performs significantly better than DDIS (blue).

Figure 12: We find that in nullspace (a) and overall (b), FunBCS (orange) has significantly less error (b) on our discontinuous Poisson problem discussed in Sec. 3, as compared to DDIS (blue) validating our theoretical results.

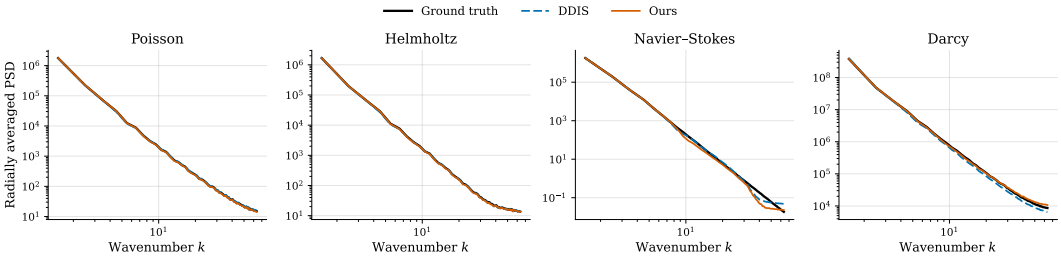


Figure 13: Radially averaged power spectral density of the Navier–Stokes reconstructions against the ground truth, alongside three other PDEs. Above $k \approx 8$, DDIS retains substantially more power than our method and exceeds the ground-truth spectrum over part of this range, which is what drives its lower E_s in Table 1.

We observe that at in Fig. 13, at the crossover at $k \approx 8$, DDIS retains considerably more power than we do, and this is what produces its lower E_s. However, the coherence of both methods is near zero over exactly this range, so the extra power DDIS retains is not aligned with the true modes.

Comparing both curves against the ground truth sharpens this. Our reconstructions fall below the true spectrum in this range, whereas DDIS rises above it over part of it, so DDIS is adding high-frequency detail that the true field does not contain and that does not match the true modes. Because this content is uncorrelated with the ground truth, it adds to the squared error rather than reducing it, which is consistent with our lower relative ℓ_2, as shrinking unrecoverable modes toward zero is the better choice under an ℓ_2 objective, even though it costs spectral realism. We therefore report E_s alongside coherence, since on this benchmark a lower E_s reflects plausible but spurious high-frequency texture rather than recovered structure.

I Additional Experiments

I.1 Visualizing Reconstructed Fields & Solutions

In this section we provide additional visualizations of the reconstructed coefficient fields, together with the solution fields estimated from the same sparse observations, on Poisson and Helmholtz (Fig. 16) and on Navier–Stokes and Darcy (Fig. 17).

I.2 Box Plots of Main Results

Figure 15 shows the distribution of the errors in Table 1 over the 100 test samples.

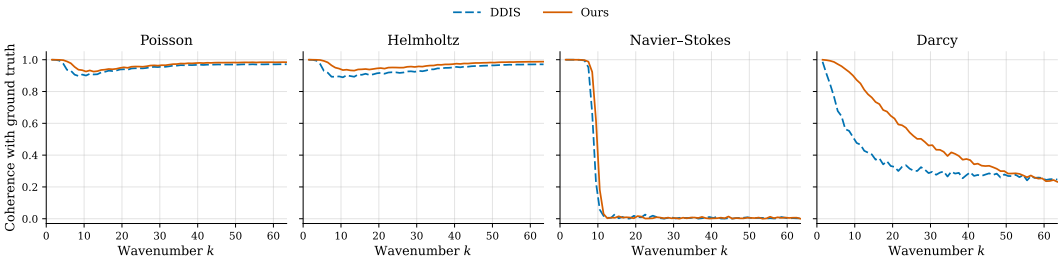


Figure 14: Per-wavenumber coherence with the ground truth on four PDEs. Beyond the same $k \approx 8$ crossover, coherence is near zero for *both* methods, indicating that neither recovers the true high-wavenumber modes. Together with Fig. 13, this suggests the E_s gap reflects prior-consistent texture in unidentified modes rather than recovered information.

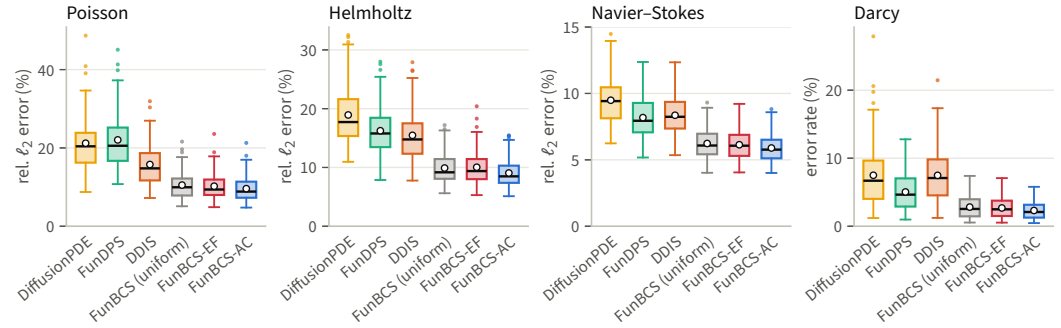


Figure 15: Per-sample errors of the recovered coefficient field on the 100 test samples of Table 1: relative ℓ_2 error on Poisson, Helmholtz and Navier–Stokes, and pixel error rate on Darcy. Each box spans the interquartile range, with the median as a line and the mean reported in Table 1 as a white dot. Wiskers extend to 1.5 times the interquartile range, and samples beyond them are drawn individually.

I.3 Study of Discontinous Poisson's Equation

In order to validate Prop. 3, and the distinction between DDIS and FunBCS more broadly, we perform an experiment on discontinuous Poisson following our setting in Sec. 3 except we use a neural operator surrogate; learned prior; compute $\bar{\boldsymbol{a}}_{D,t_{k-1}}$ by averaging 100 samples $\boldsymbol{a}^{(D)}_{t_{k-1}}$. We consider $\Omega = (-32, 32)$, and $z_- = -2, z_+ = 2$. Note that we do not configure an input optimization scheduler, instead using 10 reverse sampling steps each with 30 input optimization steps. We compute $\boldsymbol{P}_{\ker \boldsymbol{A}}$ with respect to the finite difference approximation $\boldsymbol{G}$. As expected, per Fig. 12a, FunBCS obtains significantly less error than DDIS in nullspace; per Fig. 12b, FunBCS thus obtains significantly less error overall.

J Connection with Preconditioned Output Space Optimization

As discussed in App. C, input optimization is equivalent to preconditioned output space optimization. We discuss this further below. We consider the setting of Sec. 3. Firstly, we note that output space optimization fails because the unobserved component of the measurement consistency gradient is 0:

Proposition 6. *Let* $\mathcal{L}(\boldsymbol{a}) := \frac{\|\boldsymbol{A}\boldsymbol{a} - \boldsymbol{u}_{\text{obs}}\|_2^2}{2\gamma^2}$ *for a* $\gamma > 0$. *Then,*

$$\nabla_{\boldsymbol{a}} \mathcal{L}(\boldsymbol{a}) = \frac{1}{\gamma^2} \boldsymbol{A}^T (\boldsymbol{A}\boldsymbol{a} - \boldsymbol{u}_{\text{obs}}) \tag{32}$$

and thus

$$\boldsymbol{P}_{\ker \boldsymbol{A}} \nabla_{\boldsymbol{a}} \mathcal{L}(\boldsymbol{a}) = 0 \tag{33}$$

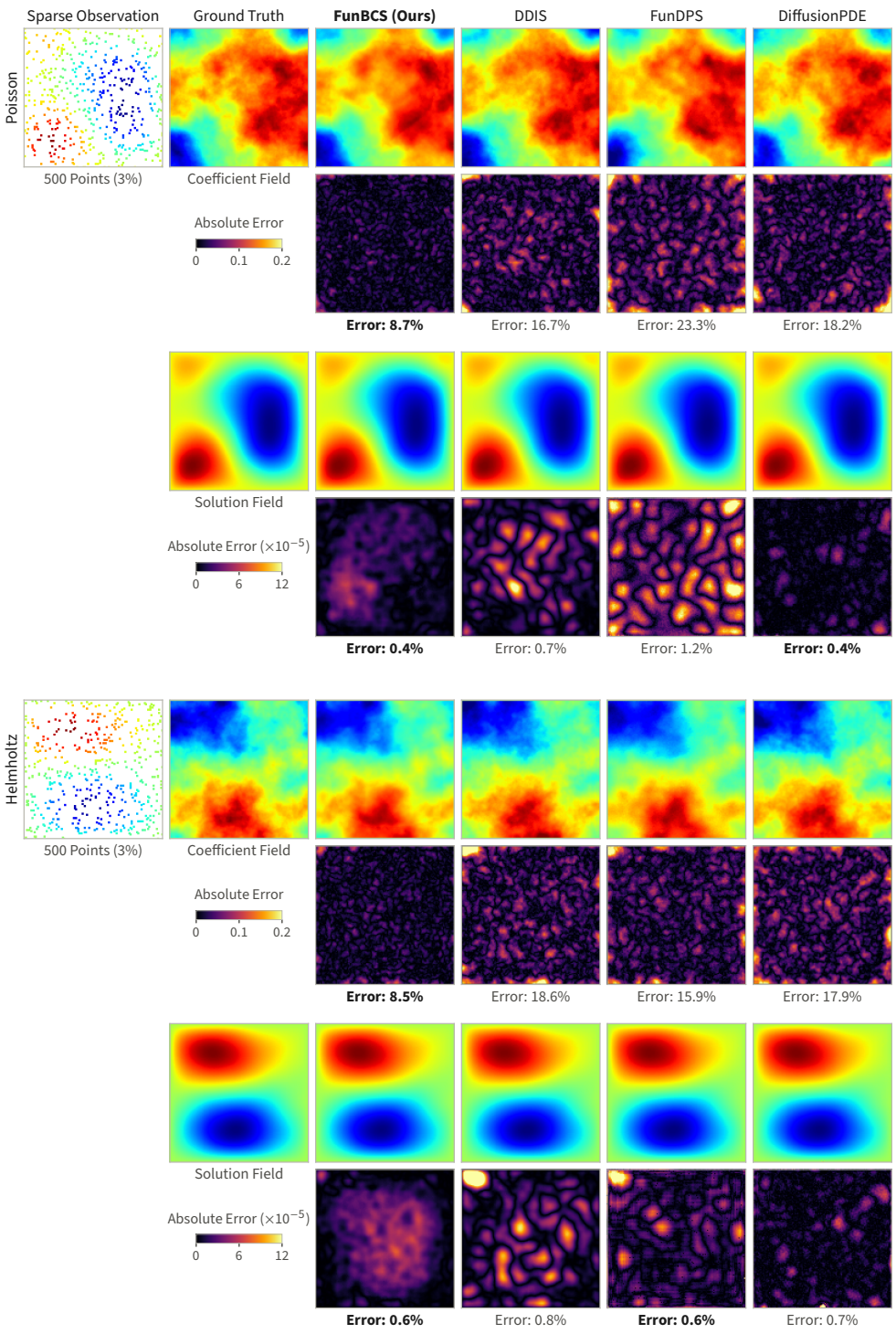


Figure 16: Reconstructions on one test sample of Poisson and of Helmholtz. For each PDE, the upper block shows the coefficient field and the lower block the solution field estimated from the same sparse observations, each with its absolute error to the ground truth underneath and its error on that sample. Every method here was run with their published default test-time compute budget, and FunBCS was ran with efficiency mode, which is $1.7\times$, $1.4\times$, and $26.4\times$ faster than DDIS, FunDPS, and DiffusionPDE, respectively. FunBCS and DDIS obtain the solution field by applying the neural operator to their recovered coefficient field, whereas FunDPS and DiffusionPDE sample both fields jointly.

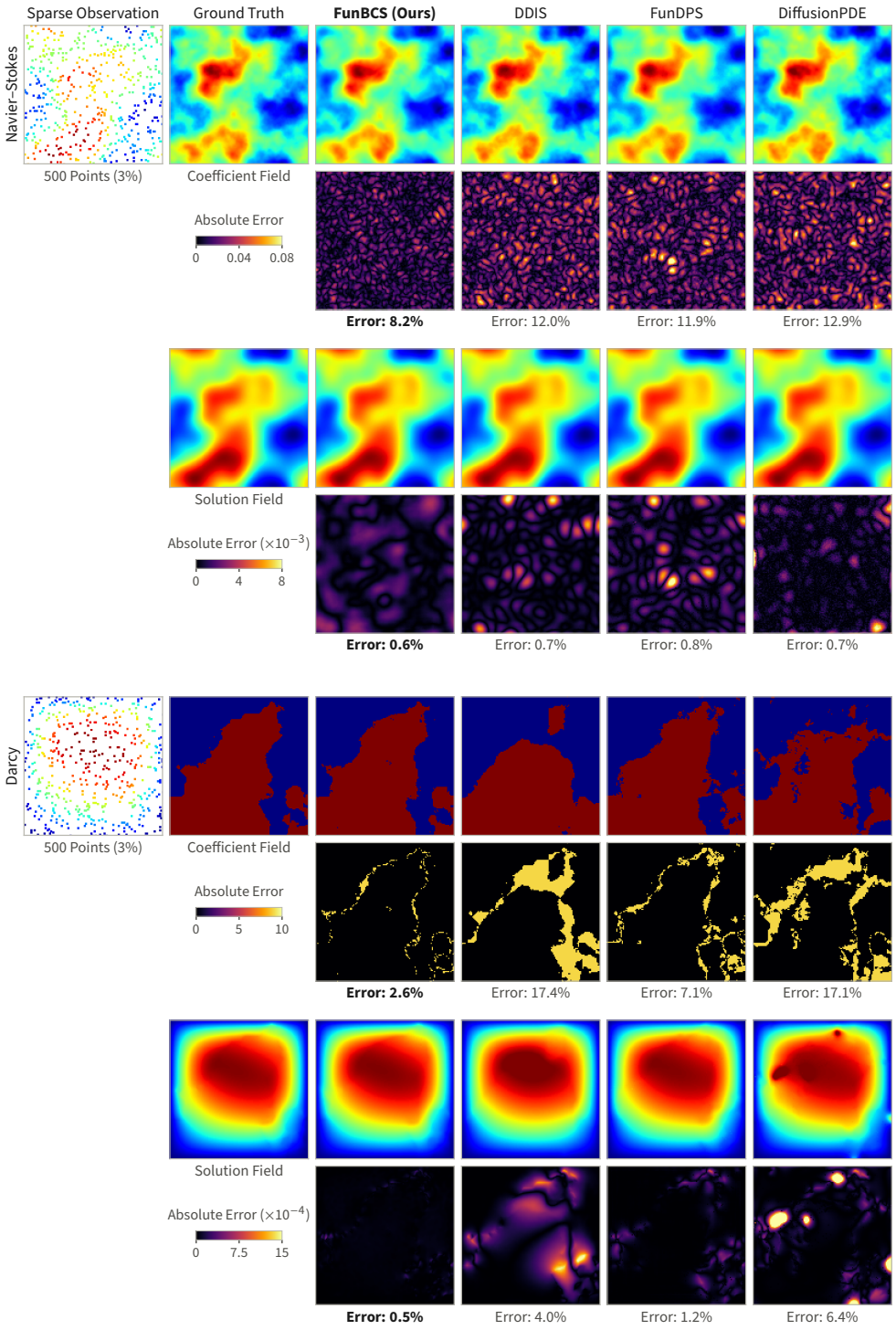


Figure 17: Reconstructions on one test sample of Navier–Stokes and of Darcy, arranged as in Fig. 16. On Darcy, the error of the coefficient field is the pixel error rate.

Proof: See App. K.9.

Consequently, an output space optimization procedure which leverages the measurement consistency gradient, unchanged, will not update the nullspace component. In particular, output space optimization can be viewed in continuous time as computing $\dot{\boldsymbol{a}} = -\nabla_{\boldsymbol{a}}\mathcal{L}(\boldsymbol{a})$, and consequently $\boldsymbol{P}_{\ker \boldsymbol{A}}\dot{\boldsymbol{a}} = 0$. However, one can leverage a *preconditioner* to couple the observed and unobserved components. Then, changing the observed component in output space by minimizing the measurement consistency loss will also change the unobserved component. In particular, our approach as outlined in Sec. 4 is equivalent to performing preconditioned output space optimization:

Proposition 7. *Assume $\boldsymbol{A}\boldsymbol{d} \neq 0$ and $\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d} \neq 0$. Fix diffusion time t_k and consider the exact denoiser $D_{\boldsymbol{\theta}}(\cdot, t_k) = D^*(\cdot, t_k)$. Let $\boldsymbol{v}(s)$ be defined per the gradient flow in Eq.* (12)*. Let $\boldsymbol{a}(s) := D^*(\boldsymbol{v}(s), t_k)$. Then, we have that:*

$$\frac{d}{ds}\boldsymbol{a}(s) = -P(v(s), t_k)\nabla_{\boldsymbol{a}}\mathcal{L}(\boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{a}(s)}, \tag{34}$$

where

$$\boldsymbol{P}(\boldsymbol{v}, t_k) := \nabla_{\boldsymbol{v}} D^*(\boldsymbol{v}, t_k)\nabla_{\boldsymbol{v}} D^*(\boldsymbol{v}, t_k)^\top. \tag{35}$$

Consequently,

$$\boldsymbol{P}_{\ker \boldsymbol{A}}\frac{d}{ds}\boldsymbol{a}(s) \neq 0. \tag{36}$$

Proof: See App. K.10.

Thus, optimizing over the input is equivalent to preconditioned output space optimization, which can update the unobserved components. We note that the recent approach of (Shi et al., 2026b) also incorporates a preconditioner, but does not motivate it rigorously and requires extra computation; our approach is instead motivated from the perspective of PDE inverse problems and leverages the diffusion denoiser.

K PROOFS

Consider the setting of App. B. Fix diffusion times $t_k > t_{k-1}$ with $\sigma_{t_k} \geq \sigma_{t_0} > 0$. Throughout, we consider the setting of Sec. 3. In proofs, $k_{t_k}(\cdot)$ denotes the function defined by $k_{t_k}(\boldsymbol{v}) := \mathbb{P}(\boldsymbol{a}_0 = \boldsymbol{a}_+ \mid \boldsymbol{a}_{t_k} = \boldsymbol{v})$, and $w(s) := k_{t_k}(\boldsymbol{v}(s))$. Here, $k_{t_k}(\boldsymbol{a}_{t_k})$ means evaluation at the forward random vector $\boldsymbol{a}_{t_k}$, while $k_{t_k}(\boldsymbol{a}_{t_k}^{(D)})$ and $k_{t_k}(\boldsymbol{a}_{t_k}^{(B)})$ mean evaluation of this same function at the algorithmic states. We show this regular conditional probability is well-defined in Lemma 4.

K.1 HELPFUL LEMMAS

Lemma 1 (Conditional laws after re-noising)**.** *The update in Eq.* (6) *satisfies*

$$\boldsymbol{a}_{t_{k-1}}^{(D)} \mid \boldsymbol{a}_{t_k}^{(D)}, \boldsymbol{u}_{\text{obs}} \sim \mathcal{N}\left(\bar{\boldsymbol{a}}_{\text{D},t_{k-1}}, \left(\frac{\boldsymbol{A}^\top\boldsymbol{A}}{\gamma^2} + \frac{\boldsymbol{I}}{\sigma_{t_k}^2}\right)^{-1} + \sigma_{t_{k-1}}^2\boldsymbol{K}\right), \tag{37}$$

where $\bar{\boldsymbol{a}}_{\text{D},t_{k-1}}$ is given by Lemma 2, and under the exact denoiser $D^*(\cdot, t_k)$, the update in Eq. (13) *satisfies:*

$$\boldsymbol{a}_{t_{k-1}}^{(B)} \mid \boldsymbol{a}_{t_k}^{(B)}, \boldsymbol{u}_{\text{obs}} \sim \mathcal{N}(D^*(\boldsymbol{v}(s), t_k), \sigma_{t_{k-1}}^2\boldsymbol{K}). \tag{38}$$

Proof. Conditional on $\boldsymbol{a}_{t_k}^{(D)}, \boldsymbol{u}_{\text{obs}}$, Lemma 2 gives

$$\tilde{\boldsymbol{a}}_0 \sim \mathcal{N}(\bar{\boldsymbol{a}}_{\text{D},t_{k-1}}, \boldsymbol{\Sigma}_{\text{D},t_k}), \qquad \boldsymbol{\varepsilon}_k^{(D)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K}), \qquad \tilde{\boldsymbol{a}}_0 \perp\!\!\!\perp \boldsymbol{\varepsilon}_k^{(D)}. \tag{39}$$

Therefore

$$\sigma_{t_{k-1}}\boldsymbol{\varepsilon}_k^{(D)} \sim \mathcal{N}(\boldsymbol{0}, \sigma_{t_{k-1}}^2\boldsymbol{K}), \qquad \tilde{\boldsymbol{a}}_0 + \sigma_{t_{k-1}}\boldsymbol{\varepsilon}_k^{(D)} \sim \mathcal{N}(\bar{\boldsymbol{a}}_{\text{D},t_{k-1}}, \boldsymbol{\Sigma}_{\text{D},t_k} + \sigma_{t_{k-1}}^2\boldsymbol{K}). \tag{40}$$

proving Eq. (37).

For Eq. (38), Lemma 6 determines $\boldsymbol{v}(s)$ uniquely for fixed $\boldsymbol{a}_{t_k}^{(B)}, \boldsymbol{u}_{\text{obs}}$. Thus $D^*(\boldsymbol{v}(s), t_k) \in \mathbb{R}^n$ is fixed under this conditioning, and

$$\boldsymbol{\varepsilon}_k^{(B)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{K}) \quad \Longrightarrow \quad D^*(\boldsymbol{v}(s), t_k) + \sigma_{t_{k-1}} \boldsymbol{\varepsilon}_k^{(B)} \sim \mathcal{N}(D^*(\boldsymbol{v}(s), t_k), \sigma_{t_{k-1}}^2 \boldsymbol{K}). \tag{41}$$

This proves Lemma 1. □

Lemma 2 (DDIS mean and covariance)**.** *The density in Eq.* (5)*, given* $\mathcal{A}_h = \boldsymbol{A}$*, is that of* $\mathcal{N}(\bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}}, \boldsymbol{\Sigma}_{\mathrm{D},t_k})$*, where*

$$\bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} = \left(\frac{\boldsymbol{A}^\top \boldsymbol{A}}{\gamma^2} + \frac{\boldsymbol{I}}{\sigma_{t_k}^2} \right)^{-1} \left(\frac{\boldsymbol{A}^\top \boldsymbol{u}_{\text{obs}}}{\gamma^2} + \frac{D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)}, t_k)}{\sigma_{t_k}^2} \right) \tag{42}$$

and

$$\boldsymbol{\Sigma}_{\mathrm{D},t_k} := \left(\frac{\boldsymbol{A}^\top \boldsymbol{A}}{\gamma^2} + \frac{\boldsymbol{I}}{\sigma_{t_k}^2} \right)^{-1}. \tag{43}$$

Proof. Fix $t_k > 0$, $\boldsymbol{a}_{t_k}^{(D)} \in \mathbb{R}^n$, and $\boldsymbol{u}_{\text{obs}} \in \mathbb{R}^m$. Throughout the proof, $\boldsymbol{a}_0 \in \mathbb{R}^n$ is the variable in the density. DDIS approximates the denoising conditional density by

$$q_D(\boldsymbol{a}_0 \mid \boldsymbol{a}_{t_k}^{(D)}) = \frac{1}{(2\pi\sigma_{t_k}^2)^{n/2}} \exp\left\{ -\frac{\|\boldsymbol{a}_0 - D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)}, t_k)\|^2}{2\sigma_{t_k}^2} \right\}. \tag{44}$$

The observation likelihood is

$$p(\boldsymbol{u}_{\text{obs}} \mid \boldsymbol{a}_0) = \frac{1}{(2\pi\gamma^2)^{m/2}} \exp\left\{ -\frac{\|\boldsymbol{A}\boldsymbol{a}_0 - \boldsymbol{u}_{\text{obs}}\|^2}{2\gamma^2} \right\}. \tag{45}$$

DDIS defines the following density

$$q_{D,t_k}(\boldsymbol{a}_0 \mid \boldsymbol{a}_{t_k}^{(D)}, \boldsymbol{u}_{\text{obs}}) := \frac{p(\boldsymbol{u}_{\text{obs}} \mid \boldsymbol{a}_0)\, q_D(\boldsymbol{a}_0 \mid \boldsymbol{a}_{t_k}^{(D)})}{\displaystyle\int_{\mathbb{R}^n} p(\boldsymbol{u}_{\text{obs}} \mid \boldsymbol{x})\, q_D(\boldsymbol{x} \mid \boldsymbol{a}_{t_k}^{(D)})\, d\boldsymbol{x}}. \tag{46}$$

The denominator is strictly positive because its integrand is everywhere positive. It is finite because $p(\boldsymbol{u}_{\text{obs}} \mid \boldsymbol{x}) \le (2\pi\gamma^2)^{-m/2}$ and $q_D(\cdot \mid \boldsymbol{a}_{t_k}^{(D)})$ integrates to one. Thus Eq. (46) is well-defined.

Substituting Eqs. (44) and (45) into Eq. (46) gives

$$q_{D,t_k}(\boldsymbol{a}_0 \mid \boldsymbol{a}_{t_k}^{(D)}, \boldsymbol{u}_{\text{obs}}) = \frac{\dfrac{\exp\left\{ -\frac{\|\boldsymbol{A}\boldsymbol{a}_0 - \boldsymbol{u}_{\text{obs}}\|^2}{2\gamma^2} - \frac{\|\boldsymbol{a}_0 - D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)}, t_k)\|^2}{2\sigma_{t_k}^2} \right\}}{(2\pi\gamma^2)^{m/2}(2\pi\sigma_{t_k}^2)^{n/2}}}{\dfrac{\int_{\mathbb{R}^n} \exp\left\{ -\frac{\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{u}_{\text{obs}}\|^2}{2\gamma^2} - \frac{\|\boldsymbol{x} - D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)}, t_k)\|^2}{2\sigma_{t_k}^2} \right\} d\boldsymbol{x}}{(2\pi\gamma^2)^{m/2}(2\pi\sigma_{t_k}^2)^{n/2}}}. \tag{47}$$

We next complete the square in the exponent. The matrix $\boldsymbol{A}^\top \boldsymbol{A}/\gamma^2 + \boldsymbol{I}/\sigma_{t_k}^2$ is symmetric, and for every nonzero $\boldsymbol{x} \in \mathbb{R}^n$,

$$\boldsymbol{x}^\top \left(\frac{\boldsymbol{A}^\top \boldsymbol{A}}{\gamma^2} + \frac{\boldsymbol{I}}{\sigma_{t_k}^2} \right) \boldsymbol{x} = \frac{\|\boldsymbol{A}\boldsymbol{x}\|^2}{\gamma^2} + \frac{\|\boldsymbol{x}\|^2}{\sigma_{t_k}^2} > 0. \tag{48}$$

Hence it is positive definite and invertible, so $\boldsymbol{\Sigma}_{\mathrm{D},t_k}$ in Lemma 2 is well-defined and positive definite. By definition, we have

$$\boldsymbol{\Sigma}_{\mathrm{D},t_k}^{-1} \bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} = \frac{\boldsymbol{A}^\top \boldsymbol{u}_{\text{obs}}}{\gamma^2} + \frac{D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)}, t_k)}{\sigma_{t_k}^2}. \tag{49}$$

Expanding the two squared norms and using Eq. (49) yields

$$\begin{aligned}
&\frac{\|\boldsymbol{A}\boldsymbol{a}_0-\boldsymbol{u}_{\text{obs}}\|^2}{2\gamma^2}+\frac{\|\boldsymbol{a}_0-D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)},t_k)\|^2}{2\sigma_{t_k}^2}\\
&=\frac{1}{2\gamma^2}(\boldsymbol{A}\boldsymbol{a}_0-\boldsymbol{u}_{\text{obs}})^\top(\boldsymbol{A}\boldsymbol{a}_0-\boldsymbol{u}_{\text{obs}})\\
&\quad+\frac{1}{2\sigma_{t_k}^2}(\boldsymbol{a}_0-D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)},t_k))^\top(\boldsymbol{a}_0-D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)},t_k))\\
&=\frac{1}{2\gamma^2}\left(\boldsymbol{a}_0^\top\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{a}_0-2\boldsymbol{a}_0^\top\boldsymbol{A}^\top\boldsymbol{u}_{\text{obs}}+\|\boldsymbol{u}_{\text{obs}}\|^2\right)\\
&\quad+\frac{1}{2\sigma_{t_k}^2}\left(\boldsymbol{a}_0^\top\boldsymbol{a}_0-2\boldsymbol{a}_0^\top D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)},t_k)+\|D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)},t_k)\|^2\right)\\
&=\frac{1}{2}\boldsymbol{a}_0^\top\left(\frac{\boldsymbol{A}^\top\boldsymbol{A}}{\gamma^2}+\frac{\boldsymbol{I}}{\sigma_{t_k}^2}\right)\boldsymbol{a}_0\\
&\quad-\boldsymbol{a}_0^\top\left(\frac{\boldsymbol{A}^\top\boldsymbol{u}_{\text{obs}}}{\gamma^2}+\frac{D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)},t_k)}{\sigma_{t_k}^2}\right)+\frac{\|\boldsymbol{u}_{\text{obs}}\|^2}{2\gamma^2}+\frac{\|D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)},t_k)\|^2}{2\sigma_{t_k}^2}\\
&=\frac{1}{2}\boldsymbol{a}_0^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}\boldsymbol{a}_0-\boldsymbol{a}_0^\top\left(\frac{\boldsymbol{A}^\top\boldsymbol{u}_{\text{obs}}}{\gamma^2}+\frac{D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)},t_k)}{\sigma_{t_k}^2}\right)+\frac{\|\boldsymbol{u}_{\text{obs}}\|^2}{2\gamma^2}+\frac{\|D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)},t_k)\|^2}{2\sigma_{t_k}^2}\\
&=\frac{1}{2}\boldsymbol{a}_0^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}\boldsymbol{a}_0-\boldsymbol{a}_0^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}\bar{\boldsymbol{a}}_{\text{D},t_{k-1}}+\frac{\|\boldsymbol{u}_{\text{obs}}\|^2}{2\gamma^2}+\frac{\|D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)},t_k)\|^2}{2\sigma_{t_k}^2}.
\end{aligned}\tag{50}$$

Symmetry of $\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}$ gives

$$\begin{aligned}
&\frac{1}{2}(\boldsymbol{a}_0-\bar{\boldsymbol{a}}_{\text{D},t_{k-1}})^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}(\boldsymbol{a}_0-\bar{\boldsymbol{a}}_{\text{D},t_{k-1}})\\
&=\frac{1}{2}(\boldsymbol{a}_0^\top-\bar{\boldsymbol{a}}_{\text{D},t_{k-1}}^\top)\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}(\boldsymbol{a}_0-\bar{\boldsymbol{a}}_{\text{D},t_{k-1}})\\
&=\frac{1}{2}\Big(\boldsymbol{a}_0^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}\boldsymbol{a}_0-\boldsymbol{a}_0^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}\bar{\boldsymbol{a}}_{\text{D},t_{k-1}}-\bar{\boldsymbol{a}}_{\text{D},t_{k-1}}^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}\boldsymbol{a}_0+\bar{\boldsymbol{a}}_{\text{D},t_{k-1}}^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}\bar{\boldsymbol{a}}_{\text{D},t_{k-1}}\Big)\\
&=\frac{1}{2}\Big(\boldsymbol{a}_0^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}\boldsymbol{a}_0-2\boldsymbol{a}_0^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}\bar{\boldsymbol{a}}_{\text{D},t_{k-1}}+\bar{\boldsymbol{a}}_{\text{D},t_{k-1}}^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}\bar{\boldsymbol{a}}_{\text{D},t_{k-1}}\Big)\\
&=\frac{1}{2}\boldsymbol{a}_0^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}\boldsymbol{a}_0-\boldsymbol{a}_0^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}\bar{\boldsymbol{a}}_{\text{D},t_{k-1}}+\frac{1}{2}\bar{\boldsymbol{a}}_{\text{D},t_{k-1}}^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}\bar{\boldsymbol{a}}_{\text{D},t_{k-1}}.
\end{aligned}\tag{51}$$

Define

$$c_{\text{D},t_k}:=\frac{\|\boldsymbol{u}_{\text{obs}}\|^2}{2\gamma^2}+\frac{\|D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)},t_k)\|^2}{2\sigma_{t_k}^2}-\frac{1}{2}\bar{\boldsymbol{a}}_{\text{D},t_{k-1}}^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}\bar{\boldsymbol{a}}_{\text{D},t_{k-1}}.\tag{52}$$

Incorporating Eq. (51) into Eq. (50) yields

$$\frac{\|\boldsymbol{A}\boldsymbol{a}_0-\boldsymbol{u}_{\text{obs}}\|^2}{2\gamma^2}+\frac{\|\boldsymbol{a}_0-D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)},t_k)\|^2}{2\sigma_{t_k}^2}=\frac{1}{2}(\boldsymbol{a}_0-\bar{\boldsymbol{a}}_{\text{D},t_{k-1}})^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}(\boldsymbol{a}_0-\bar{\boldsymbol{a}}_{\text{D},t_{k-1}})+c_{\text{D},t_k}.\tag{53}$$

The scalar c_{D,t_k} is independent of $\boldsymbol{a}_0$. Since Eq. (53) holds for every $\boldsymbol{a}_0\in\mathbb{R}^n$, it also holds with the integration variable $\boldsymbol{x}$ in place of $\boldsymbol{a}_0$, with the same scalar c_{D,t_k}. Consequently, Eq. (47) becomes

$$\begin{aligned}
q_{D,t_k}(\boldsymbol{a}_0\mid\boldsymbol{a}_{t_k}^{(D)},\boldsymbol{u}_{\text{obs}})&=\frac{e^{-c_{\text{D},t_k}}\exp\Big\{-\frac{1}{2}(\boldsymbol{a}_0-\bar{\boldsymbol{a}}_{\text{D},t_{k-1}})^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}(\boldsymbol{a}_0-\bar{\boldsymbol{a}}_{\text{D},t_{k-1}})\Big\}}{e^{-c_{\text{D},t_k}}\displaystyle\int_{\mathbb{R}^n}\exp\Big\{-\frac{1}{2}(\boldsymbol{x}-\bar{\boldsymbol{a}}_{\text{D},t_{k-1}})^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}(\boldsymbol{x}-\bar{\boldsymbol{a}}_{\text{D},t_{k-1}})\Big\}\,d\boldsymbol{x}}\\
&=\frac{\exp\Big\{-\frac{1}{2}(\boldsymbol{a}_0-\bar{\boldsymbol{a}}_{\text{D},t_{k-1}})^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}(\boldsymbol{a}_0-\bar{\boldsymbol{a}}_{\text{D},t_{k-1}})\Big\}}{\displaystyle\int_{\mathbb{R}^n}\exp\Big\{-\frac{1}{2}(\boldsymbol{x}-\bar{\boldsymbol{a}}_{\text{D},t_{k-1}})^\top\boldsymbol{\Sigma}_{\text{D},t_k}^{-1}(\boldsymbol{x}-\bar{\boldsymbol{a}}_{\text{D},t_{k-1}})\Big\}\,d\boldsymbol{x}}.
\end{aligned}\tag{54}$$

To evaluate the remaining integral, the spectral theorem provides the symmetric positive definite square root $\boldsymbol{\Sigma}_{\mathrm{D},t_k}^{1/2}$. Define $\boldsymbol{z} = \boldsymbol{\Sigma}^{-\frac{1}{2}}(\boldsymbol{x} - \bar{\boldsymbol{a}}_{D,t_{k-1}})$, i.e., $\boldsymbol{x} = \bar{\boldsymbol{a}}_{D,t_{k-1}} + \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{z}$. The map $\boldsymbol{z} \mapsto \bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} + \boldsymbol{\Sigma}_{\mathrm{D},t_k}^{1/2}\boldsymbol{z}$ is a continuously differentiable bijection of $\mathbb{R}^n$ with continuously differentiable inverse and Jacobian determinant $(\det \boldsymbol{\Sigma}_{\mathrm{D},t_k})^{1/2} > 0$. The change of variables theorem therefore gives

$$\begin{aligned}
&\int_{\mathbb{R}^n} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}})^\top \boldsymbol{\Sigma}_{\mathrm{D},t_k}^{-1}(\boldsymbol{x} - \bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}})\right\} d\boldsymbol{x} \\
&= (\det \boldsymbol{\Sigma}_{\mathrm{D},t_k})^{1/2} \int_{\mathbb{R}^n} e^{-\|\boldsymbol{z}\|^2/2}\, d\boldsymbol{z} \\
&= (\det \boldsymbol{\Sigma}_{\mathrm{D},t_k})^{1/2}(2\pi)^{n/2}.
\end{aligned} \tag{55}$$

by Tonelli's theorem and the Gaussian integral $\int_{\mathbb{R}} e^{-r^2/2}\, dr = \sqrt{2\pi}$. Substituting Eq. (55) into Eq. (54) yields

$$q_{D,t_k}(\boldsymbol{a}_0 \mid \boldsymbol{a}_{t_k}^{(D)}, \boldsymbol{u}_{\mathrm{obs}}) = \frac{\exp\left\{-\frac{1}{2}(\boldsymbol{a}_0 - \bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}})^\top \boldsymbol{\Sigma}_{\mathrm{D},t_k}^{-1}(\boldsymbol{a}_0 - \bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}})\right\}}{(2\pi)^{n/2}(\det \boldsymbol{\Sigma}_{\mathrm{D},t_k})^{1/2}}. \tag{56}$$

This is exactly the density of $\mathcal{N}(\bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}}, \boldsymbol{\Sigma}_{\mathrm{D},t_k})$, as desired.

□

Lemma 3 (Orthogonal projection onto $\ker \boldsymbol{A}$)**.** *For every $\boldsymbol{u} \in \mathbb{R}^m$ and $\boldsymbol{a} \in \mathbb{R}^n$,*

$$\boldsymbol{P}_{\ker \boldsymbol{A}}(\boldsymbol{A}^\top \boldsymbol{u}) = \boldsymbol{0}, \qquad \|\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}\|_2 \le \|\boldsymbol{a}\|_2. \tag{57}$$

Proof. For every $\boldsymbol{u} \in \mathbb{R}^m$ and $\boldsymbol{z} \in \ker \boldsymbol{A}$,

$$\langle \boldsymbol{A}^\top \boldsymbol{u}, \boldsymbol{z}\rangle = \langle \boldsymbol{u}, \boldsymbol{A}\boldsymbol{z}\rangle = 0. \tag{58}$$

Thus $\boldsymbol{A}^\top \boldsymbol{u} \in (\ker \boldsymbol{A})^\perp$, so $\boldsymbol{P}_{\ker \boldsymbol{A}}(\boldsymbol{A}^\top \boldsymbol{u}) = \boldsymbol{0}$.

For every $\boldsymbol{a} \in \mathbb{R}^n$, the vectors $\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}$ and $\boldsymbol{a} - \boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}$ are orthogonal with $\boldsymbol{a} = \boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a} + (\boldsymbol{I} - \boldsymbol{P}_{\ker \boldsymbol{A}})\boldsymbol{a}$. Hence

$$\|\boldsymbol{a}\|_2^2 = \|\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}\|_2^2 + \|\boldsymbol{a} - \boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}\|_2^2 \ge \|\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}\|_2^2, \tag{59}$$

which gives $\|\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}\|_2 \le \|\boldsymbol{a}\|_2$ as desired. □

Lemma 4 (Exact denoiser under π_0)**.** *For every $\boldsymbol{v} \in \mathbb{R}^n$,*

$$D^*(\boldsymbol{v}, t_k) = \boldsymbol{a}_- + \mathbb{P}(\boldsymbol{a}_0 = \boldsymbol{a}_+ \mid \boldsymbol{a}_{t_k} = \boldsymbol{v})(\boldsymbol{a}_+ - \boldsymbol{a}_-), \tag{60}$$

where

$$\mathbb{P}(\boldsymbol{a}_0 = \boldsymbol{a}_+ \mid \boldsymbol{a}_{t_k} = \boldsymbol{v}) = \left(1 + \exp\left\{-\frac{(\boldsymbol{a}_+ - \boldsymbol{a}_-)^\top \boldsymbol{K}^{-1}(\boldsymbol{v} - (\boldsymbol{a}_- + \boldsymbol{a}_+)/2)}{\sigma_{t_k}^2}\right\}\right)^{-1} \tag{61}$$

is well-defined.

Proof. For $\boldsymbol{v} \in \mathbb{R}^n$, by definition of the diffusion forward process, we have that

$$p(\boldsymbol{a}_{t_k} = \boldsymbol{v} \mid \boldsymbol{a}_0 = \boldsymbol{a}_\pm) = \frac{\exp\{-(\boldsymbol{v} - \boldsymbol{a}_\pm)^\top \boldsymbol{K}^{-1}(\boldsymbol{v} - \boldsymbol{a}_\pm)/(2\sigma_{t_k}^2)\}}{(2\pi)^{n/2}\sigma_{t_k}^n (\det \boldsymbol{K})^{1/2}} > 0. \tag{62}$$

By the law of total probability,

$$p(\boldsymbol{a}_{t_k} = \boldsymbol{v}) = \tfrac{1}{2}p(\boldsymbol{a}_{t_k} = \boldsymbol{v} \mid \boldsymbol{a}_0 = \boldsymbol{a}_-) + \tfrac{1}{2}p(\boldsymbol{a}_{t_k} = \boldsymbol{v} \mid \boldsymbol{a}_0 = \boldsymbol{a}_+) > 0. \tag{63}$$

Define

$$k_{t_k}(\boldsymbol{v}) := \frac{\frac{1}{2}p(\boldsymbol{a}_{t_k} = \boldsymbol{v} \mid \boldsymbol{a}_0 = \boldsymbol{a}_+)}{p(\boldsymbol{a}_{t_k} = \boldsymbol{v})}. \tag{64}$$

This function is Borel measurable and takes values in $(0,1)$. Thus $k_{t_k}(\boldsymbol{a}_{t_k})$ is integrable and $\sigma(\boldsymbol{a}_{t_k})$-measurable. For every Borel set $B \subseteq \mathbb{R}^n$,

$$\begin{aligned}\mathbb{E}[k_{t_k}(\boldsymbol{a}_{t_k})\mathbf{1}_{\{\boldsymbol{a}_{t_k}\in B\}}] &= \int_B k_{t_k}(\boldsymbol{v})p(\boldsymbol{a}_{t_k}=\boldsymbol{v})\,d\boldsymbol{v}\\ &= \frac{1}{2}\int_B p(\boldsymbol{a}_{t_k}=\boldsymbol{v}\mid \boldsymbol{a}_0=\boldsymbol{a}_+)\,d\boldsymbol{v}\\ &= \mathbb{P}(\boldsymbol{a}_0=\boldsymbol{a}_+)\mathbb{P}(\boldsymbol{a}_{t_k}\in B|\boldsymbol{a}_0=\boldsymbol{a}_+)\\ &= \mathbb{P}(\boldsymbol{a}_0=\boldsymbol{a}_+,\boldsymbol{a}_{t_k}\in B)\\ &= \mathbb{E}[\mathbf{1}_{\{\boldsymbol{a}_0=\boldsymbol{a}_+\}}\mathbf{1}_{\{\boldsymbol{a}_{t_k}\in B\}}].\end{aligned} \tag{65}$$

Since the events $\{\boldsymbol{a}_{t_k}\in B\}$ form $\sigma(\boldsymbol{a}_{t_k})$, by the definition of conditional expectation this verifies $k_{t_k}(\boldsymbol{a}_{t_k}) = \mathbb{E}[\mathbf{1}_{\{\boldsymbol{a}_0=\boldsymbol{a}_+\}} \mid \sigma(\boldsymbol{a}_{t_k})] = \mathbb{P}(\boldsymbol{a}_0=\boldsymbol{a}_+ \mid \sigma(\boldsymbol{a}_{t_k}))$ almost surely. Define $Q(\cdot,\cdot)$ by $Q(\boldsymbol{v},C) = k_{t_k}(\boldsymbol{v})\mathbf{1}_{\{\boldsymbol{a}_+\in C\}} + (1-k_{t_k}(\boldsymbol{v}))\mathbf{1}_{\{\boldsymbol{a}_-\in C\}}$ for $\boldsymbol{v}\in\mathbb{R}^n$ and Borel set $C\subset\mathbb{R}^n$. For every fixed $\boldsymbol{v}$, the map $C\mapsto Q(\boldsymbol{v},C)$ is a probability measure on $\mathbb{R}^n$, and for every Borel set $C\subset\mathbb{R}^n$, the map $\boldsymbol{v}\mapsto Q(\boldsymbol{v},C)$ is Borel measurable. Moreover, since $\boldsymbol{a}_0$ takes values only in $\{\boldsymbol{a}_+,\boldsymbol{a}_-\}$,

$$\begin{aligned}\mathbb{E}[\mathbf{1}_{\{\boldsymbol{a}_0\in C\}} \mid \sigma(\boldsymbol{a}_{t_k})] &= \mathbf{1}_{\{\boldsymbol{a}_+\in C\}}\mathbb{E}[\mathbf{1}_{\{\boldsymbol{a}_0=\boldsymbol{a}_+\}} \mid \sigma(\boldsymbol{a}_{t_k})]\\ &\quad + \mathbf{1}_{\{\boldsymbol{a}_-\in C\}}\mathbb{E}[\mathbf{1}_{\{\boldsymbol{a}_0=\boldsymbol{a}_-\}} \mid \sigma(\boldsymbol{a}_{t_k})]\\ &= \mathbf{1}_{\{\boldsymbol{a}_+\in C\}}k_{t_k}(\boldsymbol{a}_{t_k}) + \mathbf{1}_{\{\boldsymbol{a}_-\in C\}}(1-k_{t_k}(\boldsymbol{a}_{t_k}))\\ &= Q(\boldsymbol{a}_{t_k},C)\end{aligned}$$

almost surely, since $\mathbf{1}_{\{\boldsymbol{a}_0=\boldsymbol{a}_-\}} = 1-\mathbf{1}_{\{\boldsymbol{a}_0=\boldsymbol{a}_+\}}$. Hence Q is a regular conditional distribution of $\boldsymbol{a}_0$ given $\boldsymbol{a}_{t_k}$. In particular, $\mathbb{P}(\boldsymbol{a}_0=\boldsymbol{a}_+ \mid \boldsymbol{a}_{t_k}=\boldsymbol{v}) := Q(\boldsymbol{v},\{\boldsymbol{a}_+\}) = k_{t_k}(\boldsymbol{v})$ is well-defined for every $\boldsymbol{v}\in\mathbb{R}^n$.

Incorporating Eqs. (62) and (63) into Eq. (64) yields

$$\begin{aligned}k_{t_k}(\boldsymbol{v}) &= \left(1+\frac{p(\boldsymbol{a}_{t_k}=\boldsymbol{v}\mid\boldsymbol{a}_0=\boldsymbol{a}_-)}{p(\boldsymbol{a}_{t_k}=\boldsymbol{v}\mid\boldsymbol{a}_0=\boldsymbol{a}_+)}\right)^{-1}\\ &= \left(1+\exp\left\{\frac{(\boldsymbol{v}-\boldsymbol{a}_+)^\top\boldsymbol{K}^{-1}(\boldsymbol{v}-\boldsymbol{a}_+)-(\boldsymbol{v}-\boldsymbol{a}_-)^\top\boldsymbol{K}^{-1}(\boldsymbol{v}-\boldsymbol{a}_-)}{2\sigma_{t_k}^2}\right\}\right)^{-1}.\end{aligned} \tag{66}$$

By symmetry of $\boldsymbol{K}^{-1}$,

$$\begin{aligned}&(\boldsymbol{v}-\boldsymbol{a}_+)^\top\boldsymbol{K}^{-1}(\boldsymbol{v}-\boldsymbol{a}_+)-(\boldsymbol{v}-\boldsymbol{a}_-)^\top\boldsymbol{K}^{-1}(\boldsymbol{v}-\boldsymbol{a}_-)\\ &= \left(\boldsymbol{v}^\top\boldsymbol{K}^{-1}\boldsymbol{v}-\boldsymbol{v}^\top\boldsymbol{K}^{-1}\boldsymbol{a}_+-\boldsymbol{a}_+^\top\boldsymbol{K}^{-1}\boldsymbol{v}+\boldsymbol{a}_+^\top\boldsymbol{K}^{-1}\boldsymbol{a}_+\right)\\ &\quad -\left(\boldsymbol{v}^\top\boldsymbol{K}^{-1}\boldsymbol{v}-\boldsymbol{v}^\top\boldsymbol{K}^{-1}\boldsymbol{a}_--\boldsymbol{a}_-^\top\boldsymbol{K}^{-1}\boldsymbol{v}+\boldsymbol{a}_-^\top\boldsymbol{K}^{-1}\boldsymbol{a}_-\right)\\ &= -2\boldsymbol{v}^\top\boldsymbol{K}^{-1}\boldsymbol{a}_+ + 2\boldsymbol{v}^\top\boldsymbol{K}^{-1}\boldsymbol{a}_- + \boldsymbol{a}_+^\top\boldsymbol{K}^{-1}\boldsymbol{a}_+ - \boldsymbol{a}_-^\top\boldsymbol{K}^{-1}\boldsymbol{a}_-\\ &= -2\boldsymbol{v}^\top\boldsymbol{K}^{-1}(\boldsymbol{a}_+-\boldsymbol{a}_-) + \boldsymbol{a}_+^\top\boldsymbol{K}^{-1}\boldsymbol{a}_+ - \boldsymbol{a}_-^\top\boldsymbol{K}^{-1}\boldsymbol{a}_-\\ &= -2\boldsymbol{v}^\top\boldsymbol{K}^{-1}(\boldsymbol{a}_+-\boldsymbol{a}_-) + (\boldsymbol{a}_+-\boldsymbol{a}_-)^\top\boldsymbol{K}^{-1}(\boldsymbol{a}_++\boldsymbol{a}_-)\\ &= -2(\boldsymbol{a}_+-\boldsymbol{a}_-)^\top\boldsymbol{K}^{-1}\boldsymbol{v} + (\boldsymbol{a}_+-\boldsymbol{a}_-)^\top\boldsymbol{K}^{-1}(\boldsymbol{a}_++\boldsymbol{a}_-)\\ &= -2(\boldsymbol{a}_+-\boldsymbol{a}_-)^\top\boldsymbol{K}^{-1}\left(\boldsymbol{v}-\frac{\boldsymbol{a}_-+\boldsymbol{a}_+}{2}\right).\end{aligned} \tag{67}$$

proving Eq. (61).

Finally, since $\boldsymbol{a}_0$ takes only the values $\boldsymbol{a}_-,\boldsymbol{a}_+$,

$$\begin{aligned}D^*(\boldsymbol{v},t_k) &= \mathbb{E}[\boldsymbol{a}_0 \mid \boldsymbol{a}_{t_k}=\boldsymbol{v}]\\ &= (1-k_{t_k}(\boldsymbol{v}))\boldsymbol{a}_- + k_{t_k}(\boldsymbol{v})\boldsymbol{a}_+\\ &= \boldsymbol{a}_- + k_{t_k}(\boldsymbol{v})(\boldsymbol{a}_+-\boldsymbol{a}_-).\end{aligned} \tag{68}$$

Substitute $k_{t_k}(\boldsymbol{v}) = \mathbb{P}(\boldsymbol{a}_0=\boldsymbol{a}_+ \mid \boldsymbol{a}_{t_k}=\boldsymbol{v})$ to obtain Lemma 4 as desired. □

Lemma 5 (Strict positivity of the conditional weights)**.** *For every* $\boldsymbol{v} \in \mathbb{R}^n$,

$$0 < \mathbb{P}(\boldsymbol{a}_0 = \boldsymbol{a}_+ \mid \boldsymbol{a}_{t_k} = \boldsymbol{v}) < 1, \qquad 0 < \mathbb{P}(\boldsymbol{a}_0 = \boldsymbol{a}_- \mid \boldsymbol{a}_{t_k} = \boldsymbol{v}) < 1. \tag{69}$$

Proof. By Eq. (62), both conditional densities $p(\boldsymbol{a}_{t_k} = \boldsymbol{v} \mid \boldsymbol{a}_0 = \boldsymbol{a}_+)$ and $p(\boldsymbol{a}_{t_k} = \boldsymbol{v} \mid \boldsymbol{a}_0 = \boldsymbol{a}_-)$ are strictly positive. Thus

$$0 < \frac{p(\boldsymbol{a}_{t_k} = \boldsymbol{v} \mid \boldsymbol{a}_0 = \boldsymbol{a}_\pm)}{p(\boldsymbol{a}_{t_k} = \boldsymbol{v} \mid \boldsymbol{a}_0 = \boldsymbol{a}_+) + p(\boldsymbol{a}_{t_k} = \boldsymbol{v} \mid \boldsymbol{a}_0 = \boldsymbol{a}_-)} < 1. \tag{70}$$

By Eqs. (63) and (64), these ratios are the two conditional probabilities, yielding Eq. (69) as desired. □

Lemma 6 (Global existence and uniqueness of the gradient flow)**.** *For any current state* $\boldsymbol{a}_{t_k}^{(B)} \in \mathbb{R}^n$, *with the exact denoiser* $D_{\boldsymbol{\theta}}(\cdot, \cdot) = D^*(\cdot, \cdot)$, *the gradient flow in Eq.* (12)*, initialized by* $\boldsymbol{v}(0) = \boldsymbol{a}_{t_k}^{(B)}$, *has a solution* $\boldsymbol{v}(s)$ *that exists and is unique for all* $s \geq 0$.

Proof. Throughout, set

$$\boldsymbol{b} := \frac{\boldsymbol{K}^{-1}(\boldsymbol{a}_+ - \boldsymbol{a}_-)}{\sigma_{t_k}^2}, \qquad c := \frac{\|\boldsymbol{A}(\boldsymbol{a}_+ - \boldsymbol{a}_-)\|_2^2}{\gamma^2}, \qquad \ell_{t_k}(\boldsymbol{v}) := \boldsymbol{b}^\top \left(\boldsymbol{v} - \frac{\boldsymbol{a}_- + \boldsymbol{a}_+}{2} \right). \tag{71}$$

By Lemma 4,

$$k_{t_k}(\boldsymbol{v}) = \left(1 + e^{-\ell_{t_k}(\boldsymbol{v})}\right)^{-1}. \tag{72}$$

Since $\nabla_{\boldsymbol{v}} \ell_{t_k}(\boldsymbol{v}) = \boldsymbol{b}$, differentiation gives

$$\begin{aligned} \nabla_{\boldsymbol{v}} k_{t_k}(\boldsymbol{v}) &= -\left(1 + e^{-\ell_{t_k}(\boldsymbol{v})}\right)^{-2} \left(-e^{-\ell_{t_k}(\boldsymbol{v})} \boldsymbol{b}\right) \\ &= \frac{e^{-\ell_{t_k}(\boldsymbol{v})}}{\left(1 + e^{-\ell_{t_k}(\boldsymbol{v})}\right)^2} \boldsymbol{b}. \end{aligned}$$

Moreover,

$$k_{t_k}(\boldsymbol{v}) = \frac{1}{1 + e^{-\ell_{t_k}(\boldsymbol{v})}}, \qquad 1 - k_{t_k}(\boldsymbol{v}) = \frac{e^{-\ell_{t_k}(\boldsymbol{v})}}{1 + e^{-\ell_{t_k}(\boldsymbol{v})}}. \tag{73}$$

Therefore,

$$\nabla_{\boldsymbol{v}} k_{t_k}(\boldsymbol{v}) = k_{t_k}(\boldsymbol{v}) \big(1 - k_{t_k}(\boldsymbol{v})\big) \boldsymbol{b}. \tag{74}$$

Using $\boldsymbol{u}_{\text{obs}} = \boldsymbol{A}\boldsymbol{a}_+$ and Lemma 4,

$$\begin{aligned} \boldsymbol{A}D^*(\boldsymbol{v}, t_k) - \boldsymbol{u}_{\text{obs}} &= \boldsymbol{A}\boldsymbol{a}_- + k_{t_k}(\boldsymbol{v}) \boldsymbol{A}(\boldsymbol{a}_+ - \boldsymbol{a}_-) - \boldsymbol{A}\boldsymbol{a}_+ \\ &= -(1 - k_{t_k}(\boldsymbol{v})) \boldsymbol{A}(\boldsymbol{a}_+ - \boldsymbol{a}_-). \end{aligned} \tag{75}$$

Hence the measurement objective is

$$J_{t_k}(\boldsymbol{v}) := \frac{\|\boldsymbol{A}D^*(\boldsymbol{v}, t_k) - \boldsymbol{u}_{\text{obs}}\|_2^2}{2\gamma^2} = \frac{c}{2} \big(1 - k_{t_k}(\boldsymbol{v})\big)^2. \tag{76}$$

The chain rule and Eq. (74) yield

$$\begin{aligned} \nabla_{\boldsymbol{v}} J_{t_k}(\boldsymbol{v}) &= \frac{c}{2} 2\big(1 - k_{t_k}(\boldsymbol{v})\big) \big(-\nabla_{\boldsymbol{v}} k_{t_k}(\boldsymbol{v})\big) \\ &= -c\, k_{t_k}(\boldsymbol{v}) \big(1 - k_{t_k}(\boldsymbol{v})\big)^2 \boldsymbol{b}. \end{aligned} \tag{77}$$

Substituting the definitions of $\boldsymbol{b}$ and c into $\boldsymbol{v}'(s) = -\nabla J_{t_k}(\boldsymbol{v}(s))$ gives

$$\begin{aligned} \boldsymbol{v}'(s) = \frac{\|\boldsymbol{A}(\boldsymbol{a}_+ - \boldsymbol{a}_-)\|_2^2}{\gamma^2 \sigma_{t_k}^2} &\mathbb{P}\big(\boldsymbol{a}_0 = \boldsymbol{a}_+ \mid \boldsymbol{a}_{t_k} = \boldsymbol{v}(s)\big) \\ &\times \mathbb{P}\big(\boldsymbol{a}_0 = \boldsymbol{a}_- \mid \boldsymbol{a}_{t_k} = \boldsymbol{v}(s)\big)^2 \boldsymbol{K}^{-1}(\boldsymbol{a}_+ - \boldsymbol{a}_-), \\ \boldsymbol{v}(0) = \boldsymbol{a}_{t_k}^{(B)}. & \end{aligned} \tag{78}$$

It remains to show that the vector field $-\nabla J_{t_k}$ is globally Lipschitz. To compute the Hessian, fix $i, j \in \{1, \dots, n\}$. By Eq. (77),

$$\frac{\partial J_{t_k}}{\partial v_i}(\boldsymbol{v}) = -c\, b_i\, k_{t_k}(\boldsymbol{v})\big(1 - k_{t_k}(\boldsymbol{v})\big)^2. \tag{79}$$

Since b_i and c are constant with respect to $\boldsymbol{v}$, the product rule gives

$$\begin{aligned}\frac{\partial}{\partial v_j}\frac{\partial J_{t_k}}{\partial v_i}(\boldsymbol{v}) &= -c\, b_i \frac{\partial}{\partial v_j}\left[k_{t_k}(\boldsymbol{v})\big(1 - k_{t_k}(\boldsymbol{v})\big)^2\right] \\ &= -c\, b_i \left[\frac{\partial k_{t_k}}{\partial v_j}(\boldsymbol{v})\big(1 - k_{t_k}(\boldsymbol{v})\big)^2 + k_{t_k}(\boldsymbol{v})\frac{\partial}{\partial v_j}\big(1 - k_{t_k}(\boldsymbol{v})\big)^2\right].\end{aligned}$$

By the chain rule,

$$\frac{\partial}{\partial v_j}\big(1 - k_{t_k}(\boldsymbol{v})\big)^2 = -2\big(1 - k_{t_k}(\boldsymbol{v})\big)\frac{\partial k_{t_k}}{\partial v_j}(\boldsymbol{v}). \tag{80}$$

Substituting this identity gives

$$\begin{aligned}\frac{\partial}{\partial v_j}\frac{\partial J_{t_k}}{\partial v_i}(\boldsymbol{v}) &= -c\, b_i \left[\big(1 - k_{t_k}(\boldsymbol{v})\big)^2 - 2k_{t_k}(\boldsymbol{v})\big(1 - k_{t_k}(\boldsymbol{v})\big)\right]\frac{\partial k_{t_k}}{\partial v_j}(\boldsymbol{v}) \\ &= -c\, b_i\big(1 - k_{t_k}(\boldsymbol{v})\big)\big(1 - 3k_{t_k}(\boldsymbol{v})\big)\frac{\partial k_{t_k}}{\partial v_j}(\boldsymbol{v}),\end{aligned}$$

By Eq. (74),

$$\frac{\partial k_{t_k}}{\partial v_j}(\boldsymbol{v}) = k_{t_k}(\boldsymbol{v})\big(1 - k_{t_k}(\boldsymbol{v})\big)b_j. \tag{81}$$

Therefore,

$$\begin{aligned}\frac{\partial}{\partial v_j}\frac{\partial J_{t_k}}{\partial v_i}(\boldsymbol{v}) &= -c\, b_i\big(1 - k_{t_k}(\boldsymbol{v})\big)\big(1 - 3k_{t_k}(\boldsymbol{v})\big)k_{t_k}(\boldsymbol{v})\big(1 - k_{t_k}(\boldsymbol{v})\big)b_j \\ &= -c\, k_{t_k}(\boldsymbol{v})\big(1 - k_{t_k}(\boldsymbol{v})\big)^2\big(1 - 3k_{t_k}(\boldsymbol{v})\big)b_i b_j.\end{aligned}$$

Since $(\boldsymbol{b}\boldsymbol{b}^\top)_{ij} = b_i b_j$, we obtain

$$\nabla^2_{\boldsymbol{v}} J_{t_k}(\boldsymbol{v}) = -c\, k_{t_k}(\boldsymbol{v})\big(1 - k_{t_k}(\boldsymbol{v})\big)^2\big(1 - 3k_{t_k}(\boldsymbol{v})\big)\boldsymbol{b}\boldsymbol{b}^\top. \tag{82}$$

By Lemma 5, $0 < k_{t_k}(\boldsymbol{v}) < 1$. Hence

$$k_{t_k}(\boldsymbol{v}) \le 1, \qquad \big(1 - k_{t_k}(\boldsymbol{v})\big)^2 \le 1, \qquad |1 - 3k_{t_k}(\boldsymbol{v})| \le 2. \tag{83}$$

We also have

$$\|\boldsymbol{b}\boldsymbol{b}^\top\|_{\mathrm{op}} \le \|\boldsymbol{b}\|_2^2. \tag{84}$$

since for every $\boldsymbol{x} \in \mathbb{R}^n$ with $\|\boldsymbol{x}\|_2 = 1$, Cauchy–Schwarz gives

$$\begin{aligned}\|\boldsymbol{b}\boldsymbol{b}^\top\boldsymbol{x}\|_2 &= \|\boldsymbol{b}\|_2|\boldsymbol{b}^\top\boldsymbol{x}| \\ &\le \|\boldsymbol{b}\|_2^2\|\boldsymbol{x}\|_2 = \|\boldsymbol{b}\|_2^2.\end{aligned}$$

Thus, using Eq. (82),

$$\begin{aligned}\|\nabla^2_{\boldsymbol{v}} J_{t_k}(\boldsymbol{v})\|_{\mathrm{op}} &\le 2c\|\boldsymbol{b}\|_2^2 \\ &= \frac{2\|\boldsymbol{A}(\boldsymbol{a}_+ - \boldsymbol{a}_-)\|_2^2\|\boldsymbol{K}^{-1}(\boldsymbol{a}_+ - \boldsymbol{a}_-)\|_2^2}{\gamma^2\sigma_{t_k}^4} =: L_{t_k} < \infty\end{aligned} \tag{85}$$

for every $\boldsymbol{v} \in \mathbb{R}^n$.

Now let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ and define

$$\boldsymbol{g}(r) := \nabla J_{t_k}\big(\boldsymbol{y} + r(\boldsymbol{x} - \boldsymbol{y})\big), \qquad r \in [0, 1]. \tag{86}$$

Since J_{t_k} is twice continuously differentiable, the chain rule gives

$$\boldsymbol{g}'(r) = \nabla^2 J_{t_k}\big(\boldsymbol{y} + r(\boldsymbol{x}-\boldsymbol{y})\big)(\boldsymbol{x}-\boldsymbol{y}). \tag{87}$$

The fundamental theorem of calculus therefore gives

$$\begin{aligned}\nabla J_{t_k}(\boldsymbol{x}) - \nabla J_{t_k}(\boldsymbol{y}) &= \boldsymbol{g}(1) - \boldsymbol{g}(0) \\ &= \int_0^1 \nabla^2 J_{t_k}\big(\boldsymbol{y} + r(\boldsymbol{x}-\boldsymbol{y})\big)(\boldsymbol{x}-\boldsymbol{y})\, dr.\end{aligned}$$

Taking norms and applying Eq. (85),

$$\begin{aligned}\|\nabla J_{t_k}(\boldsymbol{x}) - \nabla J_{t_k}(\boldsymbol{y})\|_2 &\le \int_0^1 \big\|\nabla^2 J_{t_k}\big(\boldsymbol{y} + r(\boldsymbol{x}-\boldsymbol{y})\big)\big\|_{\mathrm{op}} \|\boldsymbol{x}-\boldsymbol{y}\|_2\, dr \\ &\le \int_0^1 L_{t_k}\|\boldsymbol{x}-\boldsymbol{y}\|_2\, dr \\ &= L_{t_k}\|\boldsymbol{x}-\boldsymbol{y}\|_2.\end{aligned} \tag{88}$$

Thus $-\nabla J_{t_k}$ is globally Lipschitz on $\mathbb{R}^n$. Consequently, the global Picard–Lindelöf theorem gives a unique solution to Eq. (12), as desired.

□

Lemma 7 (Evolution of the conditional probability under the gradient flow)**.** *Suppose* $\boldsymbol{A}(\boldsymbol{a}_+ - \boldsymbol{a}_-) \neq 0$*. For the solution the gradient flow in Eq.* (12) *initialized by* $\boldsymbol{v}(0) = \boldsymbol{a}_{t_k}^{(B)}$*,* $\boldsymbol{v}(s)$*, let* $w(s) := k_{t_k}(\boldsymbol{v}(s))$*. Then,*

$$\frac{d}{ds}w(s) = \frac{\|\boldsymbol{A}(\boldsymbol{a}_+ - \boldsymbol{a}_-)\|^2 \|\boldsymbol{K}^{-1}(\boldsymbol{a}_+ - \boldsymbol{a}_-)\|^2}{\gamma^2 \sigma_{t_k}^4} w(s)^2(1-w(s))^3. \tag{89}$$

Proof. The solution $\boldsymbol{v}(s)$ exists and is unique by Lemma 6. With $\boldsymbol{b}, c$ as in Eq. (71), Eq. (77) and the definition of gradient flow give

$$\boldsymbol{v}'(s) = -\nabla J_{t_k}(\boldsymbol{v}(s)) = cw(s)(1-w(s))^2\boldsymbol{b}. \tag{90}$$

Since $k_{t_k}(\cdot)$ and $\boldsymbol{v}(\cdot)$ are continuously differentiable, the chain rule and Eq. (74) give

$$\begin{aligned}w'(s) &= \nabla_{\boldsymbol{v}} k_{t_k}(\boldsymbol{v}(s))^\top \boldsymbol{v}'(s) \\ &= [w(s)(1-w(s))\boldsymbol{b}]^\top [cw(s)(1-w(s))^2\boldsymbol{b}] \\ &= c\|\boldsymbol{b}\|^2 w(s)^2(1-w(s))^3.\end{aligned} \tag{91}$$

Substituting the definitions of $\boldsymbol{b}, c$ gives

$$w'(s) = Cw(s)^2(1-w(s))^3, \qquad C := \frac{\|\boldsymbol{A}(\boldsymbol{a}_+ - \boldsymbol{a}_-)\|^2 \|\boldsymbol{K}^{-1}(\boldsymbol{a}_+ - \boldsymbol{a}_-)\|^2}{\gamma^2 \sigma_{t_k}^4}. \tag{92}$$

The constant is positive because $\boldsymbol{A}(\boldsymbol{a}_+ - \boldsymbol{a}_-) \neq \boldsymbol{0}$, $\boldsymbol{K}$ is invertible, and $\gamma, \sigma_{t_k} > 0$; this yields Eq. (89) as desired. □

Lemma 8 (Convergence of the conditional probability and denoiser)**.** *Suppose* $\boldsymbol{A}(\boldsymbol{a}_+ - \boldsymbol{a}_-) \neq 0$*. Along the gradient flow in Eq.* (12)*, initialized by* $\boldsymbol{v}(0) = \boldsymbol{a}_{t_k}^{(B)}$*, as* $s \to \infty$*,*

$$\mathbb{P}(\boldsymbol{a}_0 = \boldsymbol{a}_+ \mid \boldsymbol{a}_{t_k} = \boldsymbol{v}(s)) \longrightarrow 1, \qquad \mathbb{P}(\boldsymbol{a}_0 = \boldsymbol{a}_- \mid \boldsymbol{a}_{t_k} = \boldsymbol{v}(s)) \longrightarrow 0, \tag{93}$$

and

$$D^*(\boldsymbol{v}(s), t_k) \longrightarrow \boldsymbol{a}_+. \tag{94}$$

Proof. Set $w(s) := k_{t_k}(\boldsymbol{v}(s))$ and take C from Eq. (92). Note that $\boldsymbol{v}(s)$ exists and is unique by Lemma 6. By Lemmas 5 and 7, $0 < w(s) < 1$ and $w'(s) > 0$ for every finite s. Thus $w(\cdot)$ is increasing and bounded above, so its limit L exists and satisfies $0 < w(0) \le L \le 1$. Suppose $L < 1$. By continuity of the polynomial in Eq. (92),

$$w'(s) \longrightarrow CL^2(1-L)^3 =: c_0 > 0. \tag{95}$$

Hence there is $S \geq 0$ such that $w'(s) \geq c_0/2$ for all $s \geq S$. The fundamental theorem of calculus then gives

$$w(s) - w(S) = \int_S^s w'(r)\, dr \geq \frac{c_0}{2}(s - S), \qquad s \geq S. \tag{96}$$

The right-hand side grows without bound as $s \to \infty$, contradicting $L < 1$. Therefore $L = 1$, which also gives $1 - w(s) \to 0$.

Finally, by Lemma 4,

$$\begin{aligned} D^*(\boldsymbol{v}(s), t_k) - \boldsymbol{a}_+ &= \boldsymbol{a}_- + w(s)(\boldsymbol{a}_+ - \boldsymbol{a}_-) - \boldsymbol{a}_+ \\ &= -(1 - w(s))(\boldsymbol{a}_+ - \boldsymbol{a}_-), \\ \|D^*(\boldsymbol{v}(s), t_k) - \boldsymbol{a}_+\| &= (1 - w(s))\|\boldsymbol{a}_+ - \boldsymbol{a}_-\| \longrightarrow 0, \end{aligned} \tag{97}$$

because $1 - w(s) \to 0$, as desired. □

Lemma 9 (DDIS preserves the denoiser's kernel component in expectation)**.** *Fix $t_k > t_{k-1}$ and any current state $\boldsymbol{a}_{t_k}^{(D)} \in \mathbb{R}^n$. The DDIS conditional mean $\bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} := \mathbb{E}[\boldsymbol{a}_{t_{k-1}}^{(D)} \mid \boldsymbol{a}_{t_k}^{(D)}, \boldsymbol{u}_{\mathrm{obs}}]$ from Lemma 1 satisfies*

$$\boldsymbol{P}_{\ker \boldsymbol{A}} \bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} = \boldsymbol{P}_{\ker \boldsymbol{A}} D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)}, t_k). \tag{98}$$

Proof. By Lemma 2,

$$\bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} = \left(\frac{\boldsymbol{A}^\top \boldsymbol{A}}{\gamma^2} + \frac{\boldsymbol{I}}{\sigma_{t_k}^2}\right)^{-1} \left(\frac{\boldsymbol{A}^\top \boldsymbol{u}_{\mathrm{obs}}}{\gamma^2} + \frac{D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)}, t_k)}{\sigma_{t_k}^2}\right). \tag{99}$$

Multiplying by the invertible matrix $\boldsymbol{A}^\top \boldsymbol{A}/\gamma^2 + \boldsymbol{I}/\sigma_{t_k}^2$ gives

$$\frac{\boldsymbol{A}^\top \boldsymbol{A} \bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}}}{\gamma^2} + \frac{\bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}}}{\sigma_{t_k}^2} = \frac{\boldsymbol{A}^\top \boldsymbol{u}_{\mathrm{obs}}}{\gamma^2} + \frac{D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)}, t_k)}{\sigma_{t_k}^2}. \tag{100}$$

Move both terms on the right to the left and factor:

$$\begin{aligned} \mathbf{0} &= \frac{\boldsymbol{A}^\top \boldsymbol{A} \bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} - \boldsymbol{A}^\top \boldsymbol{u}_{\mathrm{obs}}}{\gamma^2} + \frac{\bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} - D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)}, t_k)}{\sigma_{t_k}^2} \\ &= \frac{\boldsymbol{A}^\top (\boldsymbol{A} \bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} - \boldsymbol{u}_{\mathrm{obs}})}{\gamma^2} + \frac{\bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} - D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)}, t_k)}{\sigma_{t_k}^2}. \end{aligned} \tag{101}$$

Apply the linear map $\boldsymbol{P}_{\ker \boldsymbol{A}}$. By Lemma 3, $\boldsymbol{P}_{\ker \boldsymbol{A}} \boldsymbol{A}^\top \boldsymbol{u} = \mathbf{0}$ for every $\boldsymbol{u} \in \mathbb{R}^m$, in particular for $\boldsymbol{u} = \boldsymbol{A} \bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} - \boldsymbol{u}_{\mathrm{obs}}$. Consequently,

$$\begin{aligned} \mathbf{0} &= \frac{\boldsymbol{P}_{\ker \boldsymbol{A}} \boldsymbol{A}^\top (\boldsymbol{A} \bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} - \boldsymbol{u}_{\mathrm{obs}})}{\gamma^2} + \frac{\boldsymbol{P}_{\ker \boldsymbol{A}} \bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} - \boldsymbol{P}_{\ker \boldsymbol{A}} D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)}, t_k)}{\sigma_{t_k}^2} \\ &= \frac{\boldsymbol{P}_{\ker \boldsymbol{A}} \bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} - \boldsymbol{P}_{\ker \boldsymbol{A}} D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)}, t_k)}{\sigma_{t_k}^2}. \end{aligned} \tag{102}$$

Since $\sigma_{t_k}^2 > 0$, we have that $\boldsymbol{P}_{\ker \boldsymbol{A}} \bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} = \boldsymbol{P}_{\ker \boldsymbol{A}} D_{\boldsymbol{\theta}}(\boldsymbol{a}_{t_k}^{(D)}, t_k)$ as desired. □

K.2 Proof of Proposition 1

Proof. By Lemma 9, with $D_{\boldsymbol{\theta}}(\cdot, t_k) = D^*(\cdot, t_k)$, and linearity of $\boldsymbol{P}_{\ker \boldsymbol{A}}$,

$$\boldsymbol{P}_{\ker \boldsymbol{A}}(\bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} - \boldsymbol{a}_+) = \boldsymbol{P}_{\ker \boldsymbol{A}} \bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} - \boldsymbol{P}_{\ker \boldsymbol{A}} \boldsymbol{a}_+ \tag{103}$$

$$= \boldsymbol{P}_{\ker \boldsymbol{A}} D^*(\boldsymbol{a}_{t_k}^{(D)}, t_k) - \boldsymbol{P}_{\ker \boldsymbol{A}} \boldsymbol{a}_+ \tag{104}$$

$$= \boldsymbol{P}_{\ker \boldsymbol{A}} \left(D^*(\boldsymbol{a}_{t_k}^{(D)}, t_k) - \boldsymbol{a}_+\right). \tag{105}$$

Taking norms on both sides proves the claim. The same argument also applies when $t_{k-1} = t_0$. □

K.3 Proof of Proposition 2

Proof. Let

$$\boldsymbol{d} := \boldsymbol{a}_+ - \boldsymbol{a}_-. \tag{106}$$

By Lemma 4, $k_{t_k}(\cdot)$ is a regular conditional probability $\mathbb{P}(\boldsymbol{a}_0 = \boldsymbol{a}_+ \mid \boldsymbol{a}_{t_k} = \cdot)$ and is well-defined, and

$$D^*(\boldsymbol{v}, t_k) = \boldsymbol{a}_- + k_{t_k}(\boldsymbol{v})\boldsymbol{d}. \tag{107}$$

The orthogonal decomposition of $\boldsymbol{d}$ with respect to $\ker \boldsymbol{A}$ is

$$\boldsymbol{d} = \boldsymbol{P}_{(\ker \boldsymbol{A})^\perp}\boldsymbol{d} + \boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d}. \tag{108}$$

Substituting this identity into Eq. (107) gives

$$D^*(\boldsymbol{v}, t_k) = \boldsymbol{a}_- + k_{t_k}(\boldsymbol{v})\boldsymbol{P}_{(\ker \boldsymbol{A})^\perp}\boldsymbol{d} + k_{t_k}(\boldsymbol{v})\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d}. \tag{109}$$

Finally, applying $\boldsymbol{P}_{\ker \boldsymbol{A}}$ and using $\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{P}_{(\ker \boldsymbol{A})^\perp} = \boldsymbol{0}$ and $\boldsymbol{P}^2_{\ker \boldsymbol{A}} = \boldsymbol{P}_{\ker \boldsymbol{A}}$ yields

$$\boldsymbol{P}_{\ker \boldsymbol{A}} D^*(\boldsymbol{v}, t_k) = \boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}_- + k_{t_k}(\boldsymbol{v})\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d}, \tag{110}$$

as desired. □

K.4 Proof of Corollary 1

Proof. Note that the probability is well-defined by Lemma 4. Let

$$\boldsymbol{d} := \boldsymbol{a}_+ - \boldsymbol{a}_-, \qquad w_D := k_{t_k}(\boldsymbol{a}^{(D)}_{t_k}). \tag{111}$$

By Lemma 4,

$$D^*(\boldsymbol{a}^{(D)}_{t_k}, t_k) - \boldsymbol{a}_+ = -(1 - w_D)\boldsymbol{d}. \tag{112}$$

Incorporating this into Prop. 1, we have that:

$$\begin{aligned} \left\|\boldsymbol{P}_{\ker \boldsymbol{A}}(\bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} - \boldsymbol{a}_+)\right\|_2 &= \left\|\boldsymbol{P}_{\ker \boldsymbol{A}}\left(D^*(\boldsymbol{a}^{(D)}_{t_k}, t_k) - \boldsymbol{a}_+\right)\right\|_2 \\ &= (1 - w_D)\|\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d}\|_2. \end{aligned} \tag{113}$$

By Lemma 3, we then have

$$\|\bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} - \boldsymbol{a}_+\|_2 \geq \left\|\boldsymbol{P}_{\ker \boldsymbol{A}}(\bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} - \boldsymbol{a}_+)\right\|_2. \tag{114}$$

Hence

$$\|\bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} - \boldsymbol{a}_+\|_2 \geq (1 - w_D)\|\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d}\|_2. \tag{115}$$

Finally, by Lemma 5, $1 - w_D > 0$, and by assumption $\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d} \neq \boldsymbol{0}$. Therefore,

$$\|\bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} - \boldsymbol{a}_+\|_2 \geq (1 - w_D)\|\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d}\|_2 > 0, \tag{116}$$

which proves Cor. 1, as desired. □

K.5 Proof of Proposition 3

Proof. Under the exact denoiser $D_{\boldsymbol{\theta}}(\cdot, t_k) = D^*(\cdot, t_k)$, Lemma 1 gives, for every finite $s \geq 0$,

$$\mathbb{E}\left[\boldsymbol{a}^{(B)}_{t_{k-1}} \,\middle|\, \boldsymbol{a}^{(B)}_{t_k}, \boldsymbol{u}_{\mathrm{obs}}\right] = D^*(\boldsymbol{v}(s), t_k). \tag{117}$$

Let

$$w_S(s) := k_{t_k}(\boldsymbol{v}(s)), \qquad w_D := k_{t_k}(\boldsymbol{a}^{(D)}_{t_k}), \qquad \boldsymbol{d} := \boldsymbol{a}_+ - \boldsymbol{a}_-. \tag{118}$$

By Lemma 4,

$$\begin{aligned} D^*(\boldsymbol{v}(s), t_k) - \boldsymbol{a}_+ &= \boldsymbol{a}_- + w_S(s)\boldsymbol{d} - \boldsymbol{a}_+ \\ &= -(1 - w_S(s))\boldsymbol{d}. \end{aligned} \tag{119}$$

Therefore

$$\|\boldsymbol{P}_{\ker \boldsymbol{A}}(D^*(\boldsymbol{v}(s), t_k) - \boldsymbol{a}_+)\|_2 = (1 - w_S(s))\|\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d}\|_2. \tag{120}$$

By Lemmas 5 and 7, $w_S'(s) > 0$ for every finite $s \geq 0$. Since $\boldsymbol{v}(0) = \boldsymbol{a}_{t_k}^{(B)}$, we have $w_S(s) > w_S(0) = k_{t_k}(\boldsymbol{a}_{t_k}^{(B)})$ for every $s > 0$. Applying Eq. (120) at s and at 0 therefore gives

$$\|\boldsymbol{P}_{\ker \boldsymbol{A}}(D^*(\boldsymbol{v}(s), t_k) - \boldsymbol{a}_+)\|_2 \leq \|\boldsymbol{P}_{\ker \boldsymbol{A}}(D^*(\boldsymbol{a}_{t_k}^{(B)}, t_k) - \boldsymbol{a}_+)\|_2, \qquad s > 0. \tag{121}$$

On the other hand, by Lemma 9,

$$\boldsymbol{P}_{\ker \boldsymbol{A}} \bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} = \boldsymbol{P}_{\ker \boldsymbol{A}} D^*(\boldsymbol{a}_{t_k}^{(D)}, t_k). \tag{122}$$

Applying Lemma 4 at $\boldsymbol{a}_{t_k}^{(D)}$ yields

$$\begin{aligned} \boldsymbol{P}_{\ker \boldsymbol{A}}(\bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} - \boldsymbol{a}_+) &= \boldsymbol{P}_{\ker \boldsymbol{A}}\big(D^*(\boldsymbol{a}_{t_k}^{(D)}, t_k) - \boldsymbol{a}_+\big) \\ &= -(1 - w_D)\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d}. \end{aligned} \tag{123}$$

Hence

$$\|\boldsymbol{P}_{\ker \boldsymbol{A}}(\bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} - \boldsymbol{a}_+)\|_2 = (1 - w_D)\|\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d}\|_2. \tag{124}$$

By Lemma 5, $0 < w_D < 1$, so $1 - w_D > 0$. By Lemma 8,

$$w_S(s) \longrightarrow 1 \qquad \text{as } s \to \infty. \tag{125}$$

Therefore there exists $s^* > 0$ such that, for every $s \geq s^*$,

$$1 - w_S(s) \leq 1 - w_D. \tag{126}$$

Multiplying by $\|\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d}\|_2$ and using Eqs. (120) and (124) gives

$$\|\boldsymbol{P}_{\ker \boldsymbol{A}}(D^*(\boldsymbol{v}(s), t_k) - \boldsymbol{a}_+)\|_2 \leq \|\boldsymbol{P}_{\ker \boldsymbol{A}}(\bar{\boldsymbol{a}}_{\mathrm{D},t_{k-1}} - \boldsymbol{a}_+)\|_2, \tag{127}$$

as desired. □

K.6 Proof of Theorem 1

Proof. By Lemma 1, for every finite s,

$$\pi_s^{(B)} = \mathcal{N}\Big(D^*(\boldsymbol{v}(s), t_k), \sigma_{t_{k-1}}^2 \boldsymbol{K}\Big). \tag{128}$$

By Lemma 8, $D^*(\boldsymbol{v}(s), t_k) \to \boldsymbol{a}_+$ as $s \to \infty$. Hence, for every $\boldsymbol{\xi} \in \mathbb{R}^n$, the characteristic function of $\pi_s^{(B)}$ satisfies

$$\begin{aligned} \widehat{\pi}_s^{(B)}(\boldsymbol{\xi}) &= \exp\left(i\boldsymbol{\xi}^\top D^*(\boldsymbol{v}(s), t_k) - \frac{1}{2}\sigma_{t_{k-1}}^2 \boldsymbol{\xi}^\top \boldsymbol{K}\boldsymbol{\xi}\right) \\ &\longrightarrow \exp\left(i\boldsymbol{\xi}^\top \boldsymbol{a}_+ - \frac{1}{2}\sigma_{t_{k-1}}^2 \boldsymbol{\xi}^\top \boldsymbol{K}\boldsymbol{\xi}\right). \end{aligned} \tag{129}$$

The limit is the characteristic function of $\mathcal{N}(\boldsymbol{a}_+, \sigma_{t_{k-1}}^2 \boldsymbol{K})$ and is continuous at $\boldsymbol{\xi} = \boldsymbol{0}$. Therefore, Lévy's continuity theorem gives

$$\pi_s^{(B)} \Rightarrow \mathcal{N}(\boldsymbol{a}_+, \sigma_{t_{k-1}}^2 \boldsymbol{K}) \qquad \text{as } s \to \infty. \tag{130}$$

Finally, when $t_{k-1} = t_0$, for every $\boldsymbol{\xi} \in \mathbb{R}^n$,

$$\exp\left(i\boldsymbol{\xi}^\top \boldsymbol{a}_+ - \frac{1}{2}\sigma_{t_0}^2 \boldsymbol{\xi}^\top \boldsymbol{K}\boldsymbol{\xi}\right) \longrightarrow \exp(i\boldsymbol{\xi}^\top \boldsymbol{a}_+) \qquad \text{as } \sigma_{t_0} \to 0. \tag{131}$$

The limit is the characteristic function of $\delta_{\boldsymbol{a}_+}$, so another application of Lévy's continuity theorem yields

$$\mathcal{N}(\boldsymbol{a}_+, \sigma_{t_0}^2 \boldsymbol{K}) \Rightarrow \delta_{\boldsymbol{a}_+}. \tag{132}$$

This proves Thm. 1, as desired. □

K.7 Proof of Proposition 4

Proof. Let $\boldsymbol{d} := \boldsymbol{a}_+ - \boldsymbol{a}_-$, $\boldsymbol{m} := (\boldsymbol{a}_- + \boldsymbol{a}_+)/2$, $Q := \boldsymbol{d}^\top \boldsymbol{K}^{-1}\boldsymbol{d}$, and

$$C := \frac{\|\boldsymbol{A}\boldsymbol{d}\|_2^2 \|\boldsymbol{K}^{-1}\boldsymbol{d}\|_2^2}{\gamma^2}.$$

Since $\boldsymbol{A}\boldsymbol{d} \neq 0$ and $\boldsymbol{K}$ is positive definite, we have $Q, C > 0$.

Fix a realization of $\boldsymbol{a}_{t_{k+1}}^{(B)}$ satisfying $k_{t_{k+1}}(\boldsymbol{a}_{t_{k+1}}^{(B)}) \leq 1/2 - \delta$, and set $p := k_{t_{k+1}}(\boldsymbol{a}_{t_{k+1}}^{(B)})$. By Lemma 4,

$$D^*(\boldsymbol{a}_{t_{k+1}}^{(B)}, t_{k+1}) = \boldsymbol{a}_- + p\boldsymbol{d} \tag{133}$$

$$= \boldsymbol{m} - \left(\frac{1}{2} - p\right)\boldsymbol{d}. \tag{134}$$

Therefore, since no optimization is performed at t_{k+1},

$$\boldsymbol{a}_{t_k}^{(B)} = \boldsymbol{m} - \left(\frac{1}{2} - p\right)\boldsymbol{d} + \sigma_{t_k}\boldsymbol{\varepsilon}_k^{(B)}, \qquad \boldsymbol{\varepsilon}_k^{(B)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K}).$$

Conditionally on $\boldsymbol{a}_{t_{k+1}}^{(B)}$, $\boldsymbol{d}^\top \boldsymbol{K}^{-1}\boldsymbol{\varepsilon}_k^{(B)} \sim \mathcal{N}(0, Q)$, since

$$\mathrm{Var}\left(\boldsymbol{d}^\top \boldsymbol{K}^{-1}\boldsymbol{\varepsilon}_k^{(B)}\right) = \boldsymbol{d}^\top \boldsymbol{K}^{-1}\boldsymbol{K}\boldsymbol{K}^{-1}\boldsymbol{d} = Q.$$

Thus, with $Z := \boldsymbol{d}^\top \boldsymbol{K}^{-1}\boldsymbol{\varepsilon}_k^{(B)}/\sqrt{Q}$, we have $Z \sim \mathcal{N}(0,1)$ and

$$\boldsymbol{d}^\top \boldsymbol{K}^{-1}(\boldsymbol{a}_{t_k}^{(B)} - \boldsymbol{m}) = -Q\left(\frac{1}{2} - p\right) + \sigma_{t_k}\sqrt{Q}\, Z. \tag{135}$$

Define $E := \{Z \leq \delta\sqrt{Q}/(2\sigma_{t_k})\}$. Since $1/2 - p \geq \delta$, on E,

$$\boldsymbol{d}^\top \boldsymbol{K}^{-1}(\boldsymbol{a}_{t_k}^{(B)} - \boldsymbol{m}) \leq -Q\delta + \sigma_{t_k}\sqrt{Q}\frac{\delta\sqrt{Q}}{2\sigma_{t_k}} \tag{136}$$

$$= -\frac{Q\delta}{2}. \tag{137}$$

For $x > 0$, the standard Gaussian Chernoff bound gives

$$\Pr(Z > x) = \Pr(e^{xZ} > e^{x^2}) \leq e^{-x^2}\mathbb{E}[e^{xZ}] = e^{-x^2/2}.$$

Taking $x = \delta\sqrt{Q}/(2\sigma_{t_k})$ therefore yields

$$\Pr\left(E \,\middle|\, \boldsymbol{a}_{t_{k+1}}^{(B)}\right) \geq 1 - \exp\left(-\frac{Q\delta^2}{8\sigma_{t_k}^2}\right). \tag{138}$$

Now set $w(s) := k_{t_k}(\boldsymbol{v}_{t_k}(s))$. Since $\boldsymbol{v}_{t_k}(0) = \boldsymbol{a}_{t_k}^{(B)}$, Lemma 4 and Eq. (137) give, on E,

$$w(0) = \left(1 + \exp\left\{-\frac{\boldsymbol{d}^\top \boldsymbol{K}^{-1}(\boldsymbol{a}_{t_k}^{(B)} - \boldsymbol{m})}{\sigma_{t_k}^2}\right\}\right)^{-1} \tag{139}$$

$$\leq \left(1 + \exp\left\{\frac{Q\delta}{2\sigma_{t_k}^2}\right\}\right)^{-1} < \frac{1}{2}. \tag{140}$$

By Lemma 7,

$$w'(s) = \frac{C}{\sigma_{t_k}^4} w(s)^2(1 - w(s))^3. \tag{141}$$

By Lemma 5, $0 < w(s) < 1$ for every finite s, so $w'(s) > 0$. By Lemma 8, $w(s) \to 1$ as $s \to \infty$. Hence, on E, $\tau_{t_k} < \infty$ and, by continuity, $w(\tau_{t_k}) = 1/2$.

Integrating Eq. (141) from 0 to τ_{t_k} and using $u = w(s)$ gives

$$\tau_{t_k} = \frac{\sigma_{t_k}^4}{C} \int_{w(0)}^{1/2} \frac{du}{u^2(1-u)^3} \tag{142}$$

$$\geq \frac{\sigma_{t_k}^4}{C} \int_{w(0)}^{1/2} \frac{du}{u^2} \tag{143}$$

$$= \frac{\sigma_{t_k}^4}{C} \left(\frac{1}{w(0)} - 2 \right), \tag{144}$$

where the inequality follows from $(1-u)^{-3} \geq 1$ for $u \in (0, 1/2]$. By Eq. (140),

$$\frac{1}{w(0)} \geq 1 + \exp\left(\frac{Q\delta}{2\sigma_{t_k}^2} \right),$$

so, on E,

$$\tau_{t_k} \geq \frac{\sigma_{t_k}^4}{C} \left[\exp\left(\frac{Q\delta}{2\sigma_{t_k}^2} \right) - 1 \right].$$

Consequently, by Eq. (138),

$$\Pr\left(\tau_{t_k} \geq \frac{\sigma_{t_k}^4}{C} \left[\exp\left(\frac{Q\delta}{2\sigma_{t_k}^2} \right) - 1 \right] \middle| \, \boldsymbol{a}_{t_{k+1}}^{(B)} \right) \tag{145}$$

$$\geq 1 - \exp\left(-\frac{Q\delta^2}{8\sigma_{t_k}^2} \right), \tag{146}$$

as desired. □

K.8 Proof of Proposition 5

Proof. Let $\boldsymbol{d} := \boldsymbol{a}_+ - \boldsymbol{a}_-$, $\boldsymbol{m} := (\boldsymbol{a}_- + \boldsymbol{a}_+)/2$, and $C := \|\boldsymbol{A}\boldsymbol{d}\|_2^2 \|\boldsymbol{K}^{-1}\boldsymbol{d}\|_2^2/\gamma^2$. Since $\boldsymbol{A}\boldsymbol{d} \neq 0$, $\boldsymbol{K}$ is positive definite, and $\gamma > 0$, we have $C > 0$.

By Lemma 4, for every $\sigma > 0$,

$$k_\sigma(\boldsymbol{v}) = \left(1 + \exp\left\{ -\frac{\boldsymbol{d}^\top \boldsymbol{K}^{-1}(\boldsymbol{v} - \boldsymbol{m})}{\sigma^2} \right\} \right)^{-1}. \tag{147}$$

Set $h := \boldsymbol{d}^\top \boldsymbol{K}^{-1}(\boldsymbol{v}^\star - \boldsymbol{m})$. By assumption, $k_\sigma(\boldsymbol{v}^\star) < 1/2$ for every $\sigma > 0$. Using Eq. (147),

$$k_\sigma(\boldsymbol{v}^\star) < \frac{1}{2} \iff \exp\left(-\frac{h}{\sigma^2} \right) > 1 \iff h < 0.$$

Hence, setting $b := -h > 0$,

$$w_\sigma(0) = k_\sigma(\boldsymbol{v}^\star) = \frac{1}{1 + \exp(b/\sigma^2)} \in \left(0, \frac{1}{2} \right). \tag{148}$$

Fix $\sigma > 0$, and write $\boldsymbol{v}_\sigma(s)$ for the solution of Eq. (12) at noise level σ, initialized by $\boldsymbol{v}_\sigma(0) = \boldsymbol{v}^\star$. By Lemma 6, this solution exists uniquely for all $s \geq 0$. Let $w_\sigma(s) := k_\sigma(\boldsymbol{v}_\sigma(s))$.

By Lemma 7, with σ in place of σ_{t_k},

$$w_\sigma'(s) = \frac{C}{\sigma^4} w_\sigma(s)^2 (1 - w_\sigma(s))^3. \tag{149}$$

By Lemma 5, $0 < w_\sigma(s) < 1$ for every finite s. Since $C > 0$, Eq. (149) gives $w_\sigma'(s) > 0$, so w_σ is strictly increasing. Moreover, by Lemma 8, $w_\sigma(s) \to 1$ as $s \to \infty$. Together with Eq. (148) and continuity of w_σ, this implies $0 < \tau(\sigma) < \infty$ and

$$w_\sigma(\tau(\sigma)) = \frac{1}{2}. \tag{150}$$

Rearranging Eq. (149), integrating from 0 to $\tau(\sigma)$, and using the change of variables $u = w_\sigma(s)$ gives

$$\tau(\sigma) = \frac{\sigma^4}{C} \int_0^{\tau(\sigma)} \frac{w'_\sigma(s)}{w_\sigma(s)^2(1 - w_\sigma(s))^3}\, ds \tag{151}$$

$$= \frac{\sigma^4}{C} \int_{w_\sigma(0)}^{1/2} \frac{du}{u^2(1-u)^3}. \tag{152}$$

The change of variables is valid because w_σ is continuously differentiable and strictly increasing.

We first consider $\sigma \downarrow 0$. For $u \in (0, 1/2]$, $(1-u)^{-3} \geq 1$, so

$$\tau(\sigma) \geq \frac{\sigma^4}{C} \int_{w_\sigma(0)}^{1/2} \frac{du}{u^2} \tag{153}$$

$$= \frac{\sigma^4}{C} \left(\frac{1}{w_\sigma(0)} - 2 \right) \tag{154}$$

$$= \frac{\sigma^4}{C} \left(\exp\left(\frac{b}{\sigma^2} \right) - 1 \right), \tag{155}$$

where the last equality follows from Eq. (148). For every $x > 0$, the power series for e^x gives $e^x - 1 \geq x^3/6$. Taking $x = b/\sigma^2$ in Eq. (155),

$$\tau(\sigma) \geq \frac{\sigma^4}{C} \frac{1}{6} \left(\frac{b}{\sigma^2} \right)^3 = \frac{b^3}{6C\sigma^2} \longrightarrow \infty \qquad \text{as } \sigma \downarrow 0. \tag{156}$$

We next consider $\sigma \to \infty$. For $u \in (0, 1/2]$, $u^2(1-u)^3 \leq u^2 \leq 1/4$, so $[u^2(1-u)^3]^{-1} \geq 4$. Therefore

$$\tau(\sigma) \geq \frac{4\sigma^4}{C} \left(\frac{1}{2} - w_\sigma(0) \right). \tag{157}$$

Set $x := b/\sigma^2$. By Eq. (148),

$$\frac{1}{2} - w_\sigma(0) = \frac{e^x - 1}{2(1 + e^x)}. \tag{158}$$

For all sufficiently large σ, $0 < x \leq 1$. Then $e^x - 1 \geq x$ and $e^x \leq e$, so

$$\frac{1}{2} - w_\sigma(0) \geq \frac{x}{2(1+e)} = \frac{b}{2(1+e)\sigma^2}.$$

Combining this with Eq. (157),

$$\tau(\sigma) \geq \frac{2b}{C(1+e)} \sigma^2 \longrightarrow \infty \qquad \text{as } \sigma \to \infty. \tag{159}$$

It remains to show that τ attains its minimum. Define $H(q) := \int_q^{1/2} u^{-2}(1-u)^{-3}\, du$ for $q \in (0, 1/2)$. The fundamental theorem of calculus gives $H'(q) = -q^{-2}(1-q)^{-3}$, so H is continuous. By Eq. (148), $\sigma \mapsto w_\sigma(0)$ is continuous, and therefore Eq. (152) gives

$$\tau(\sigma) = \frac{\sigma^4}{C} H(w_\sigma(0)),$$

so τ is continuous on $(0, \infty)$.

Fix any $\bar{\sigma} > 0$ and set $M := \tau(\bar{\sigma}) + 1$. By Eqs. (156) and (159), there exist $0 < \sigma_{\text{low}} < \bar{\sigma} < \sigma_{\text{high}} < \infty$ such that

$$\tau(\sigma) > M \qquad \text{whenever } 0 < \sigma \leq \sigma_{\text{low}} \text{ or } \sigma \geq \sigma_{\text{high}}. \tag{160}$$

Since τ is continuous on the compact interval $[\sigma_{\text{low}}, \sigma_{\text{high}}]$, the extreme value theorem gives some $\sigma^\star \in [\sigma_{\text{low}}, \sigma_{\text{high}}]$ such that

$$\tau(\sigma^\star) = \min_{\sigma \in [\sigma_{\text{low}}, \sigma_{\text{high}}]} \tau(\sigma). \tag{161}$$

Since $\bar{\sigma} \in [\sigma_{\text{low}}, \sigma_{\text{high}}]$, we have $\tau(\sigma^\star) \leq \tau(\bar{\sigma}) < M$. Thus, Eq. (160) gives $\tau(\sigma) > \tau(\sigma^\star)$ for every $\sigma \notin [\sigma_{\text{low}}, \sigma_{\text{high}}]$, while minimality on the compact interval gives $\tau(\sigma) \geq \tau(\sigma^\star)$ for every $\sigma \in [\sigma_{\text{low}}, \sigma_{\text{high}}]$. Therefore,

$$\tau(\sigma^\star) = \min_{\sigma > 0} \tau(\sigma), \qquad \sigma^\star \in [\sigma_{\text{low}}, \sigma_{\text{high}}], \tag{162}$$

as desired. □

K.9 Proof of Proposition 6

Proof. Fix $\boldsymbol{a} \in \mathbb{R}^n$. For any $\boldsymbol{h} \in \mathbb{R}^n$, define $\phi(\varepsilon) := \mathcal{L}(\boldsymbol{a} + \varepsilon \boldsymbol{h})$. Then

$$\phi(\varepsilon) = \frac{1}{2\gamma^2} \|\boldsymbol{A}\boldsymbol{a} - \boldsymbol{u}_{\text{obs}} + \varepsilon \boldsymbol{A}\boldsymbol{h}\|_2^2 \tag{163}$$

$$= \frac{1}{2\gamma^2} \left(\|\boldsymbol{A}\boldsymbol{a} - \boldsymbol{u}_{\text{obs}}\|_2^2 + 2\varepsilon (\boldsymbol{A}\boldsymbol{h})^\top (\boldsymbol{A}\boldsymbol{a} - \boldsymbol{u}_{\text{obs}}) + \varepsilon^2 \|\boldsymbol{A}\boldsymbol{h}\|_2^2 \right). \tag{164}$$

Therefore,

$$\phi'(0) = \frac{1}{\gamma^2} (\boldsymbol{A}\boldsymbol{h})^\top (\boldsymbol{A}\boldsymbol{a} - \boldsymbol{u}_{\text{obs}}) \tag{165}$$

$$= \boldsymbol{h}^\top \left[\frac{1}{\gamma^2} \boldsymbol{A}^\top (\boldsymbol{A}\boldsymbol{a} - \boldsymbol{u}_{\text{obs}}) \right]. \tag{166}$$

Since

$$\phi'(0) = D_{\boldsymbol{h}} \mathcal{L}(\boldsymbol{a}) = \langle \nabla_{\boldsymbol{a}} \mathcal{L}(\boldsymbol{a}), \boldsymbol{h} \rangle = \boldsymbol{h}^\top \nabla_{\boldsymbol{a}} \mathcal{L}(\boldsymbol{a}), \tag{167}$$

for every $\boldsymbol{h} \in \mathbb{R}^n$, comparison with the expression above gives

$$\nabla_{\boldsymbol{a}} \mathcal{L}(\boldsymbol{a}) = \frac{1}{\gamma^2} \boldsymbol{A}^\top (\boldsymbol{A}\boldsymbol{a} - \boldsymbol{u}_{\text{obs}}). \tag{168}$$

which proves the first claim.

We now show that this gradient has no component in $\ker \boldsymbol{A}$. Set

$$\boldsymbol{y} := \boldsymbol{A}\boldsymbol{a} - \boldsymbol{u}_{\text{obs}} \in \mathbb{R}^m. \tag{169}$$

For every $\boldsymbol{z} \in \ker \boldsymbol{A}$,

$$\left\langle \boldsymbol{A}^\top \boldsymbol{y}, \boldsymbol{z} \right\rangle = \langle \boldsymbol{y}, \boldsymbol{A}\boldsymbol{z} \rangle = 0, \tag{170}$$

because $\boldsymbol{A}\boldsymbol{z} = 0$. Hence

$$\boldsymbol{A}^\top \boldsymbol{y} \in (\ker \boldsymbol{A})^\perp. \tag{171}$$

The orthogonal projection of a vector in $(\ker \boldsymbol{A})^\perp$ onto $\ker \boldsymbol{A}$ is zero. Consequently,

$$\boldsymbol{P}_{\ker \boldsymbol{A}} \nabla_{\boldsymbol{a}} \mathcal{L}(\boldsymbol{a}) = \frac{1}{\gamma^2} \boldsymbol{P}_{\ker \boldsymbol{A}} \boldsymbol{A}^\top (\boldsymbol{A}\boldsymbol{a} - \boldsymbol{u}_{\text{obs}}) \tag{172}$$

$$= 0, \tag{173}$$

as desired. □

K.10 Proof of Proposition 7

Proof. Let $\boldsymbol{d} := \boldsymbol{a}_+ - \boldsymbol{a}_-$, $\quad w(s) := k_{t_k}(\boldsymbol{v}(s))$. Define the Jacobian of the exact denoiser by $\boldsymbol{J}_{D^*}(\boldsymbol{v}, t_k) := \nabla_{\boldsymbol{v}} D^*(\boldsymbol{v}, t_k) \in \mathbb{R}^{n \times n}$, and define

$$\boldsymbol{P}(\boldsymbol{v}, t_k) := \boldsymbol{J}_{D^*}(\boldsymbol{v}, t_k) \boldsymbol{J}_{D^*}(\boldsymbol{v}, t_k)^\top. \tag{174}$$

Define $\mathcal{J}_{t_k}(\boldsymbol{v}) := \mathcal{L}(D^*(\boldsymbol{v}, t_k))$. The gradient flow in Eq. (12) is therefore $\boldsymbol{v}'(s) = -\nabla_{\boldsymbol{v}} \mathcal{J}_{t_k}(\boldsymbol{v})\big|_{\boldsymbol{v}=\boldsymbol{v}(s)}$. By the multivariate chain rule,

$$\nabla_{\boldsymbol{v}} \mathcal{J}_{t_k}(\boldsymbol{v}) = \boldsymbol{J}_{D^*}(\boldsymbol{v}, t_k)^\top \nabla_{\boldsymbol{a}} \mathcal{L}(\boldsymbol{a})\big|_{\boldsymbol{a}=D^*(\boldsymbol{v}, t_k)}. \tag{175}$$

Consequently,

$$\boldsymbol{v}'(s) = -\boldsymbol{J}_{D^*}(\boldsymbol{v}(s), t_k)^\top \nabla_{\boldsymbol{a}} \mathcal{L}(\boldsymbol{a})\big|_{\boldsymbol{a}=\boldsymbol{a}(s)}, \tag{176}$$

where $\boldsymbol{a}(s) := D^*(\boldsymbol{v}(s), t_k)$. Differentiating $\boldsymbol{a}(s)$ and applying the chain rule once more gives

$$\boldsymbol{a}'(s) = \boldsymbol{J}_{D^*}(\boldsymbol{v}(s), t_k) \boldsymbol{v}'(s) \tag{177}$$

$$= -\boldsymbol{J}_{D^*}(\boldsymbol{v}(s), t_k) \boldsymbol{J}_{D^*}(\boldsymbol{v}(s), t_k)^\top \nabla_{\boldsymbol{a}} \mathcal{L}(\boldsymbol{a})\big|_{\boldsymbol{a}=\boldsymbol{a}(s)} \tag{178}$$

$$= -\boldsymbol{P}(\boldsymbol{v}(s), t_k) \nabla_{\boldsymbol{a}} \mathcal{L}(\boldsymbol{a})\big|_{\boldsymbol{a}=\boldsymbol{a}(s)}. \tag{179}$$

as desired.

We next show explicitly that the preconditioner in Eq. (174) can introduce a nonzero measurement nullspace component. By Lemma 4, we have that

$$D^*(\boldsymbol{v}, t_k) = \boldsymbol{a}_- + k_{t_k}(\boldsymbol{v})\boldsymbol{d}. \tag{180}$$

Moreover, by Lemma 4, we have that

$$k_{t_k}(\boldsymbol{v}) = \left(1 + \exp\left\{-\frac{\boldsymbol{d}^\top \boldsymbol{K}^{-1}\left(\boldsymbol{v} - \frac{\boldsymbol{a}_- + \boldsymbol{a}_+}{2}\right)}{\sigma_{t_k}^2}\right\}\right)^{-1}. \tag{181}$$

Thus,

$$\nabla_{\boldsymbol{v}} k_{t_k}(\boldsymbol{v}) = \frac{k_{t_k}(\boldsymbol{v})(1 - k_{t_k}(\boldsymbol{v}))}{\sigma_{t_k}^2}\boldsymbol{K}^{-1}\boldsymbol{d}. \tag{182}$$

Differentiating Eq. (180) with respect to $\boldsymbol{v}$ therefore gives

$$\boldsymbol{J}_{D^*}(\boldsymbol{v}, t_k) = \boldsymbol{d}\left(\nabla_{\boldsymbol{v}} k_{t_k}(\boldsymbol{v})\right)^\top. \tag{183}$$

Substituting Eq. (182) into Eq. (183) and then into Eq. (174) yields

$$\begin{aligned}
\boldsymbol{P}(\boldsymbol{v}, t_k) &= \boldsymbol{d}\left(\nabla_{\boldsymbol{v}} k_{t_k}(\boldsymbol{v})\right)^\top \nabla_{\boldsymbol{v}} k_{t_k}(\boldsymbol{v})\boldsymbol{d}^\top && (184)\\
&= \|\nabla_{\boldsymbol{v}} k_{t_k}(\boldsymbol{v})\|_2^2\, \boldsymbol{d}\boldsymbol{d}^\top && (185)\\
&= \frac{k_{t_k}(\boldsymbol{v})^2(1 - k_{t_k}(\boldsymbol{v}))^2}{\sigma_{t_k}^4}\|\boldsymbol{K}^{-1}\boldsymbol{d}\|_2^2 \boldsymbol{d}\boldsymbol{d}^\top. && (186)
\end{aligned}$$

We now evaluate the output space optimization along $\boldsymbol{v}(s)$. By Eq. (180), $\boldsymbol{a}(s) = \boldsymbol{a}_- + w(s)\boldsymbol{d}$. Since $\boldsymbol{u}_{\text{obs}} = \boldsymbol{A}\boldsymbol{a}_+$ and $\boldsymbol{a}_+ = \boldsymbol{a}_- + \boldsymbol{d}$,

$$\begin{aligned}
\boldsymbol{A}\boldsymbol{a}(s) - \boldsymbol{u}_{\text{obs}} &= \boldsymbol{A}\boldsymbol{a}_- + w(s)\boldsymbol{A}\boldsymbol{d} - \boldsymbol{A}\boldsymbol{a}_+ && (187)\\
&= -(1 - w(s))\boldsymbol{A}\boldsymbol{d}. && (188)
\end{aligned}$$

By Prop. 6,

$$\begin{aligned}
\nabla_{\boldsymbol{a}}\mathcal{L}(\boldsymbol{a})\big|_{\boldsymbol{a}=\boldsymbol{a}(s)} &= \frac{1}{\gamma^2}\boldsymbol{A}^\top\left(\boldsymbol{A}\boldsymbol{a}(s) - \boldsymbol{u}_{\text{obs}}\right) && (189)\\
&= -\frac{1 - w(s)}{\gamma^2}\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{d}. && (190)
\end{aligned}$$

Using Eq. (186) and Eq. (190) in Eq. (179) gives

$$\begin{aligned}
\boldsymbol{a}'(s) &= \frac{w(s)^2(1 - w(s))^3}{\gamma^2\sigma_{t_k}^4}\|\boldsymbol{K}^{-1}\boldsymbol{d}\|_2^2 \boldsymbol{d}\boldsymbol{d}^\top\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{d} && (191)\\
&= \frac{\|\boldsymbol{A}\boldsymbol{d}\|_2^2\|\boldsymbol{K}^{-1}\boldsymbol{d}\|_2^2}{\gamma^2\sigma_{t_k}^4}w(s)^2(1 - w(s))^3\boldsymbol{d}, && (192)
\end{aligned}$$

since $\boldsymbol{d}^\top\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{d} = \|\boldsymbol{A}\boldsymbol{d}\|_2^2$. Applying $\boldsymbol{P}_{\ker \boldsymbol{A}}$ to Eq. (192) yields

$$\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}'(s) = \frac{\|\boldsymbol{A}\boldsymbol{d}\|_2^2\|\boldsymbol{K}^{-1}\boldsymbol{d}\|_2^2}{\gamma^2\sigma_{t_k}^4}w(s)^2(1 - w(s))^3\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d}. \tag{193}$$

For every finite $s \geq 0$, we have that $w(s) \in (0, 1)$ by Lemma 5, and furthermore $\boldsymbol{A}\boldsymbol{d} \neq 0$ and $\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d} \neq 0$ by assumption. Since $\boldsymbol{K}$ is positive definite, it is invertible, and $\boldsymbol{d} \neq 0$ implies $\|\boldsymbol{K}^{-1}\boldsymbol{d}\|_2^2 > 0$. Together with $\gamma > 0$ and $\sigma_{t_k} > 0$, every scalar factor multiplying $\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d}$ in Eq. (193) is strictly positive. Therefore, $\boldsymbol{P}_{\ker \boldsymbol{A}}\frac{d}{ds}\boldsymbol{a}(s) = \boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}'(s) \neq 0$ for every finite $s \geq 0$, as desired. □

K.11 Statement and Proof of Proposition 8

We first state the proposition:

Proposition 8. *Assume $0 < z_- < z_+ < 1$ without loss of generality, and $1 \le m \le n$. Let $\boldsymbol{A} = \boldsymbol{OG}$ and $\boldsymbol{P}_{\ker \boldsymbol{A}}$ be the orthogonal projection matrix of a vector onto $\ker \boldsymbol{A}$. Under a centered finite difference discretization of $\mathcal{G}$ as $\boldsymbol{G}$ with $n > \max\{\frac{1}{z_-}, \frac{1}{1-z_+}, \frac{2}{z_+ - z_-}\}$, if $\boldsymbol{O}$ samples m points uniformly at random, without replacement, from $\boldsymbol{u}$, then $\boldsymbol{A}\boldsymbol{a}_+ \neq \boldsymbol{A}\boldsymbol{a}_-$ and with probability at least $1 - (m/n)^4$, $\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}_+ \neq \boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{a}_-$.*

Discussion: The condition on n ensures that the resolution is fine enough that there are some grid points between the locations of z_- and z_+. Additionally, we ensure that z_- and z_+ are not at the end of the grid.

We then provide the proof, which leverages the classic result by Thomas (1949):

Proof. Let

$$h := \frac{1}{n+1}, \qquad x_i := ih, \qquad i = 0, \dots, n+1, \tag{194}$$

so that $x_0 = 0$ and $x_{n+1} = 1$. Under the centered finite difference discretization of $u''(\cdot) = a(\cdot)$ on $(0, 1)$ with homogeneous Dirichlet boundary conditions, define $\boldsymbol{L} \in \mathbb{R}^{n \times n}$ by

$$(\boldsymbol{L}\boldsymbol{u})_i := \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2}, \qquad i = 1, \dots, n, \tag{195}$$

where $u_0 = u_{n+1} = 0$. The corresponding discrete solution operator is $\boldsymbol{G} = \boldsymbol{L}^{-1}$.

We first identify the indices on which $\boldsymbol{d} := \boldsymbol{a}_+ - \boldsymbol{a}_-$ is nonzero. Define

$$p := \min\{i \in \{1, \dots, n\} : x_i \ge z_-\}, \qquad q := \max\{i \in \{1, \dots, n\} : x_i < z_+\}. \tag{196}$$

The assumption $n > \max\left\{\frac{1}{z_-}, \frac{1}{1-z_+}, \frac{2}{z_+ - z_-}\right\}$ implies $h = \frac{1}{n+1} < \frac{1}{n} < z_-$, and hence $x_1 = h < z_-$, so $p \ge 2$. Similarly, $h < \frac{1}{n} < 1 - z_+$, so that $x_n = 1 - h > z_+$ and therefore $q \le n - 1$. Finally, we have that $2h < \frac{2}{n} < z_+ - z_-$ By the definition of p, $x_{p-1} < z_- \le x_p$, so that $x_{p+1} = x_{p-1} + 2h < z_- + 2h < z_+$, and thus

$$2 \le p < q \le n - 1. \tag{197}$$

Since $z_- < z_+$, the two discretized coefficient fields differ precisely at the grid points $x_i \in [z_-, z_+)$. Hence, setting $c := a^{(1)} - a^{(2)} \neq 0$, we have

$$\boldsymbol{d} = c \sum_{j=p}^{q} \boldsymbol{e}_j, \tag{198}$$

where $\boldsymbol{e}_j$ is the jth standard basis vector of $\mathbb{R}^n$.

We first show that $\boldsymbol{A}\boldsymbol{d} \neq 0$. For $i, j \in \{1, \dots, n\}$, define

$$g_{ij} := -h \min(x_i, x_j)\,(1 - \max(x_i, x_j))\,, \tag{199}$$

and let $\boldsymbol{g}_j := (g_{1j}, \dots, g_{nj})^\top$. We verify that $\boldsymbol{L}\boldsymbol{g}_j = \boldsymbol{e}_j$. Firstly, extend the definition of g_{ij} so that $g_{0j} = g_{n+1,j} = 0$. If $i < j$, then $g_{ij} = -hx_i(1 - x_j)$, which yields $\frac{g_{i-1,j} - 2g_{ij} + g_{i+1,j}}{h^2} = 0$. Similarly, if $i > j$, then $g_{ij} = -hx_j(1 - x_i)$, which then yields that $\frac{g_{i-1,j} - 2g_{ij} + g_{i+1,j}}{h^2} = 0$. When $i = j$, using $x_{j-1} = x_j - h$ and $x_{j+1} = x_j + h$,

$$g_{j-1,j} - 2g_{jj} + g_{j+1,j} = -h\Big[(x_j - h)(1 - x_j) - 2x_j(1 - x_j) + x_j(1 - x_j - h)\Big] \tag{200}$$

$$= h^2. \tag{201}$$

Consequently, $\frac{g_{j-1,j} - 2g_{jj} + g_{j+1,j}}{h^2} = 1$. Thus $\boldsymbol{L}\boldsymbol{g}_j = \boldsymbol{e}_j$ for every j, so $\boldsymbol{G} = \boldsymbol{L}^{-1} = [\boldsymbol{g}_1, \dots, \boldsymbol{g}_n]$. In particular, $\boldsymbol{G}$ is symmetric, since $g_{ij} = g_{ji}$, and for every $i, j \in \{1, \dots, n\}$, $g_{ij} < 0$, because $h > 0$ and $x_i, x_j \in (0, 1)$.

It follows that, for every $i = 1, \ldots, n$, $(\boldsymbol{G}\boldsymbol{d})_i = c\sum_{j=p}^{q} g_{ij}$, and since $g_{ij} < 0, c \neq 0$, $(\boldsymbol{G}\boldsymbol{d})_i \neq 0$ for every $i = 1, ..., n$, and thus $\boldsymbol{A}\boldsymbol{d} = \boldsymbol{O}\boldsymbol{G}\boldsymbol{d} \neq 0$ as desired.

It remains to characterize $\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d}$. We have that

$$\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d} = 0 \iff \boldsymbol{d} \in \operatorname{range}(\boldsymbol{A}^T) \tag{202}$$

$$\iff \boldsymbol{d} \in \operatorname{range}(\boldsymbol{G}\boldsymbol{O}^\top) \tag{203}$$

$$\iff \boldsymbol{L}\boldsymbol{d} \in \operatorname{range}(\boldsymbol{O}^\top), \tag{204}$$

where the last equivalence follows from $\boldsymbol{L} = \boldsymbol{G}^{-1}$.

We now compute $\boldsymbol{L}\boldsymbol{d}$. Since

$$d_i = \begin{cases} c, & p \leq i \leq q, \\ 0, & \text{otherwise}, \end{cases} \tag{205}$$

and $2 \leq p < q \leq n - 1$, the centered difference gives

$$(\boldsymbol{L}\boldsymbol{d})_i = \begin{cases} \dfrac{c}{h^2}, & i = p - 1, \\ -\dfrac{c}{h^2}, & i = p, \\ -\dfrac{c}{h^2}, & i = q, \\ \dfrac{c}{h^2}, & i = q + 1, \\ 0, & \text{otherwise}. \end{cases} \tag{206}$$

Equivalently, $\boldsymbol{L}\boldsymbol{d} = \frac{c}{h^2}\left(\boldsymbol{e}_{p-1} - \boldsymbol{e}_p - \boldsymbol{e}_q + \boldsymbol{e}_{q+1}\right)$. Since $c \neq 0$ and $p < q$, its support is therefore exactly

$$B := \operatorname{supp}(\boldsymbol{L}\boldsymbol{d}) = \{p - 1, p, q, q + 1\}, \qquad |B| = 4. \tag{207}$$

Let $S \subset \{1, \ldots, n\}, \qquad |S| = m > 0$ be the set of indices sampled by $\boldsymbol{O}$. Since $\boldsymbol{O}$ is a selection matrix, $\operatorname{range}(\boldsymbol{O}^\top) = \operatorname{span}\{\boldsymbol{e}_i : i \in S\}$. Thus $\boldsymbol{L}\boldsymbol{d} \in \operatorname{range}(\boldsymbol{O}^\top) \iff B \subset S$, so that $\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d} = 0 \iff B \subset S$.

If $m < 4$, then $\mathbb{P}(B \subset S) = 0$. If $m \geq 4$, since S is chosen uniformly from the m-element subsets of $\{1, \ldots, n\}$,

$$\mathbb{P}(B \subset S) = \frac{\binom{n-4}{m-4}}{\binom{n}{m}} \tag{208}$$

$$= \frac{m(m-1)(m-2)(m-3)}{n(n-1)(n-2)(n-3)} \tag{209}$$

$$= \prod_{r=0}^{3} \frac{m - r}{n - r}. \tag{210}$$

For $r = 0, 1, 2, 3$ and $m \leq n$, $\frac{m-r}{n-r} \leq \frac{m}{n}$, since $n(m - r) \leq m(n - r) \iff mr \leq nr$. Therefore, in either case $m < 4$ or $m \geq 4$,

$$\mathbb{P}\left(\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d} = 0\right) = \mathbb{P}(B \subset S) \leq \left(\frac{m}{n}\right)^4. \tag{211}$$

Taking complements yields

$$\mathbb{P}\left(\boldsymbol{P}_{\ker \boldsymbol{A}}\boldsymbol{d} \neq 0\right) \geq 1 - \left(\frac{m}{n}\right)^4. \tag{212}$$

Together with $\boldsymbol{A}\boldsymbol{d} \neq 0$, this proves the proposition as desired. □