SBNO : Schrödinger Bridge Neural Operator for the Forward and Inverse Problems under Missing Information

Jaehyeon Park
SolverX
jaehyeon.park@solverx.ai

Inhyeok Jeong
KAIST
inhyeok.jeong@kaist.ac.kr

Noseong Park[*]
KAIST
noseong@kaist.ac.kr

Abstract

Partial differential equation (PDE) surrogate modeling often relies on either dense observations or access to governing equations/physics constraints. In practice, observations can be sparse and partial, and the underlying physics may be partially specified or unavailable. We introduce SBNO, an equation-free paired-data diffusion bridge matching framework that learns conditional operators for both forward and inverse PDE problems under partial observations. SBNO uses diffusion Schrödinger bridge matching to learn a pair of time-dependent drifts from a mask-aware endpoint coupling that preserves empirical input-solution pairing during training and supports both PDE-task directions within a single framework. At inference time, SBNO produces a deterministic point reconstruction by integrating the probability-flow ODE constructed from the learned drift pair, without inference-time observation or PDE guidance-weight tuning. On five PDE benchmarks with 5% observations, SBNO achieves superior or comparable accuracy to baselines while demonstrating lower test-instance error dispersion. Furthermore, SBNO is at least $9.7\times$ faster and reduces GPU memory by at least $4.3\times$ compared to guided-diffusion baselines.

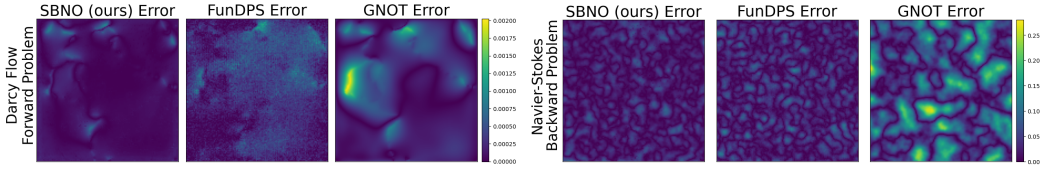


Figure 1: **Reconstruction error visualization**. We compare our method against FunDPS, a guided-diffusion baseline, and GNOT, a deterministic baseline. All error maps are visualized using a shared color scale for fair comparison, where lower values indicate better reconstruction accuracy.

1 Introduction

Partial differential equations (PDEs) are the core mathematical language for describing complex physical phenomena in a wide range of domains, including porous media flow, wave propagation, and fluid dynamics. In many scientific and engineering applications, two complementary tasks frequently arise. The forward problem involves predicting the solution field given an input field such as coefficients, sources, or initial/boundary conditions. Conversely, the inverse problem aims to infer the input field from observations of the solution. This task is often ill-posed, as it may have

[*]Corresponding Author

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

multiple plausible solutions even when the forward model is well-defined [35, 11]. Although classical numerical solvers remain the gold standard for solving PDEs due to their high accuracy [28], their computational cost can become prohibitive in scenarios requiring PDEs to be solved repeatedly, such as design optimization, control, and inverse modeling. This has led to growing interest in learning-based surrogate models in Scientific Machine Learning (SciML) [36, 34], which amortize computation by learning mappings from input fields to solution fields directly from data [30, 21, 17]. Despite their promise, a key limitation is their reliance on relatively complete observations during both training and evaluation. However, observations are often sparse in practice. Under these partial observation settings, the performance of conventional neural operators, such as Fourier Neural Operator (FNO) [17] and Deep Operator Network (DeepONet) [21], can degrade significantly [10].

Crucially, real-world scientific and engineering settings rarely provide such idealized information. Incompleteness typically stems from two main sources. First, the governing equations may be only partially specified or entirely unknown, which limits the applicability of equation-based constraints and calibrated simulators [1, 31, 26]. Second, observations are often sparse and partial due to limitations in sensor availability and incomplete domain coverage, leaving large portions of the field unobserved [24]. These two challenges frequently co-occur, making reliable PDE-based reasoning fundamentally difficult. Recent generative PDE solvers are able to address partial observations using guided diffusion [4], where observation consistency is enforced through a guidance term applied only at observed points [10, 38]. While these approaches have demonstrated promise in conditioning PDE solutions on partial observations, efficient operator learning with reliable reconstruction performance in this regime remains relatively underexplored.

To enable efficient and accurate operator learning under partial observations, we leverage diffusion bridge matching inspired by Schrödinger bridges and entropy-regularized transport [16, 3, 32]. Unlike the classical Schrödinger bridge problem, which optimizes an endpoint coupling under prescribed endpoint marginals, our method keeps the empirical endpoint coupling induced by paired PDE data fixed and learns the associated bridge dynamics. This bridge-based perspective provides a smooth diffusion path between paired endpoints, while in our method the endpoint coupling itself is prescribed by the training data rather than optimized [6]. In contrast to guided diffusion-based models, our method explicitly conditions on the observation pattern by providing the network with both the masked input field and its corresponding binary mask. This mask-aware conditioning distinguishes observed values from zero-filled missing entries and provides explicit observation-support information for forward and inverse reconstruction. In our experiments, we additionally report the dispersion of reconstruction errors across test instances, measured by the standard deviation of relative L_2 errors over the test set.

Moreover, our approach offers practical advantages for deployment. In contrast to guided diffusion-based solvers, which require inference-time guidance-weight tuning as shown in Appendix F, our method performs deterministic inference via the probability-flow ODE constructed from the learned drift pair [32]. This enables generation of a single high-fidelity reconstruction without reliance on stochastic sampling. This leads to significantly reduced memory usage and faster inference relative to existing diffusion-based PDE solvers [10, 38]. Finally, our approach is equation-free, requiring no access to the governing PDE during training or inference. Our method assumes access to fully observed input-solution pairs during training, but requires neither the governing PDE nor physics-based constraints during training or inference. We summarize the main contributions of this work as follows:

- Our method achieves better or comparable reconstruction accuracy to prior baselines while exhibiting lower error dispersion across test samples.
- Compared to diffusion-based PDE solvers, our approach is at least $9.7\times$ faster and uses at least $4.3\times$ less GPU memory at inference time.
- Our inference is tuning-free with no inference-time hyperparameter search and equation-free, requiring no access to the governing PDE during training and inference.

2 Related Works

PDE Inverse Problems PDE inverse problems aim to infer unknown coefficients, sources, or boundary conditions from observations of the state and are commonly formulated as PDE-

constrained optimization problems or within a Bayesian framework [33, 11, 9]. While classical approaches can be accurate, they are often expensive at scale. Learning-based methods have therefore been explored to amortize inference, including PINN-style formulations [37, 39, 12] and operator learning approaches [20, 22, 2, 25, 23]. Recent work [19] has also explored unified formulations that jointly address forward and inverse problems within a single framework. Nevertheless, under sparse or irregular observations, the mapping from observations to parameters is frequently ill-posed. These limitations motivate conditional generative formulations that can represent uncertainty and produce diverse solutions consistent with the same partial observations.

Function-space Generative Methods Function-space generative models address a different need than standard neural solvers. Instead of learning a single best prediction, they aim to represent a distribution over fields in an infinite-dimensional space. The Generative Adversarial Neural Operator (GANO) [29] adapts GAN [7]-style objectives to functions by combining coordinate-based decoders with global latent variables, while the Denoising Diffusion Operator [18] uses operator parameterizations to model stochasticity more explicitly and improve generalization on grids. More recently, diffusion-based formulations [10, 38] have become popular because they offer stable training and flexible conditioning and have been explored for generating PDE solutions and imputing missing regions under partial observations. Despite their strong empirical performance, many of these methods rely on stochastic sampling at inference or on hand-designed guidance mechanisms to enforce consistency with sparse observations, which can be costly or sensitive to conditioning design. This motivates bridge-based alternatives that incorporate partial observations into the endpoint construction and learn stochastic dynamics between the resulting field representations.

Diffusion Bridge and Schrödinger Bridge Schrödinger bridges enforce endpoint marginals by construction while staying close to a reference diffusion in KL divergence, providing a principled link to entropy-regularized optimal transport [5, 15]. Building on this view, recent works have proposed scalable solvers by learning the bridge dynamics from data. De Bortoli et al. [3] introduced a Diffusion Schrödinger Bridge method that alternates between forward/backward processes via Iterative Proportional Fitting. Subsequent work [32] improves stability and scalability via matching-style objectives, while Denoising Diffusion Bridge Models [40] learn bridge scores for paired distribution translation as an alternative bridge-based formulation. In our setting, DSBM provides the bridge-matching machinery. Our contribution is a PDE-specific, mask-aware endpoint coupling and its unified use for forward and inverse reconstruction.

3 Preliminary

3.1 Path Measures and a Reference Diffusion

Let $\mathbf{X}_{0:T} := (\mathbf{X}_t)_{t\in[0,T]}$ be a continuous-time stochastic process with values in $\mathbb{R}^d$. Given a set of probability measures $\mathcal{P}(\Omega)$ on path space $\Omega := C([0,T],\mathbb{R}^d)$, a *path measure* $\mathbb{P} \in \mathcal{P}(\Omega)$ denotes the probability law of the entire trajectory $\mathbf{X}_{0:T}$. Its marginal distribution at time t is denoted by $\mathbb{P}_t$, i.e., $\mathbb{P}_t$ is the law of $\mathbf{X}_t$ under $\mathbb{P}$. A distinguished reference path measure $\mathbb{Q}$ is fixed as the law induced by the diffusion SDE

$$d\mathbf{X}_t = f_t(\mathbf{X}_t)\,dt + \sigma_t\,d\mathbf{B}_t, \qquad \mathbf{X}_0 \sim \pi_0, \tag{1}$$

where $\mathbf{B}_t$ is a standard Brownian motion, $f_t : [0,T]\times\mathbb{R}^d \to \mathbb{R}^d$ is a drift, and σ_t is a scalar diffusion coefficient. Standard regularity assumptions, such as local Lipschitzness of f_t, ensure the well-posedness of (1) and the existence of the associated path measure $\mathbb{Q}$.

Given two path measures $\mathbb{P}$ and $\mathbb{Q}$ with $\mathbb{P}$ absolutely continuous with respect to $\mathbb{Q}$, the path-space Kullback-Leibler (KL) divergence is defined as:

$$\mathrm{KL}(\mathbb{P}\|\mathbb{Q}) := \begin{cases} \mathbb{E}_{\mathbb{P}}\left[\log \dfrac{d\mathbb{P}}{d\mathbb{Q}}(\mathbf{X}_{0:T})\right] & \text{if } \mathbb{P} \ll \mathbb{Q}, \\ +\infty & \text{otherwise.} \end{cases} \tag{2}$$

Crucially, if $\mathbb{P} \ll \mathbb{Q}$ is induced by an SDE with the same diffusion coefficient $\sigma_t > 0$ but a different drift $u_t(x)$, Girsanov's theorem [14] under standard Novikov-type conditions allows the KL

divergence to be expressed as the expected kinetic energy of the drift difference:

$$\mathrm{KL}(\mathbb{P}\|\mathbb{Q}) = \frac{1}{2}\mathbb{E}_{\mathbb{P}}\left[\int_0^T \|(u_t - f_t)(\mathbf{X}_t)\|^2/\sigma_t^2 dt\right]. \tag{3}$$

This formulation connects the probabilistic divergence minimization to a stochastic optimal control problem.

3.2 Schrödinger Bridge and Entropy-regularized Optimal Transport

The *Schrödinger Bridge* (SB) problem seeks to find a path measure $\mathbb{P}^*$ that minimizes the KL divergence subject to fixed initial and terminal marginal constraints π_0 and π_T as

$$\mathbb{P}^* = \underset{\mathbb{P}\in\mathcal{P}(\Omega)}{\arg\min}\, \mathrm{KL}(\mathbb{P}\|\mathbb{Q}) \quad \text{s.t.} \quad \mathbb{P}_0 = \pi_0,\ \mathbb{P}_T = \pi_T. \tag{4}$$

Whenever $\mathbb{P} \ll \mathbb{Q}$ and both measures share the same diffusion coefficient σ_t, the Girsanov representation (3) connects Eq. (4) to a stochastic optimal control formulation. The Schrödinger bridge problem can also be interpreted as an *entropic* variant of optimal transport. Among all path measures satisfying the endpoint constraints, the bridge selects the one that deviates the least from the reference dynamics $\mathbb{Q}$. Equivalently, it is the maximum-entropy solution with respect to $\mathbb{Q}$ under the same constraints. This connection becomes explicit at the level of the endpoint coupling. Let $\mathbb{Q}_{0,T}$ denote the joint law of (X_0, X_T) under the reference diffusion $\mathbb{Q}$, with joint density $q_{0,T}(x,y) = \pi_0(x)\,p(0,x;T,y)$ when $X_0 \sim \pi_0$ and $p(0,\cdot\,;T,\cdot\,)$ is the transition density. Then the optimal endpoint coupling induced by the Schrödinger bridge solves the *static* problem

$$\Pi^\star_{0,T} = \arg\min_{\Pi\in\Pi(\pi_0,\pi_T)} \mathrm{KL}(\Pi\,\|\,\mathbb{Q}_{0,T}),$$

where $\Pi(\pi_0,\pi_T)$ denotes the set of couplings with marginals π_0 and π_T. Moreover, expanding the KL shows an entropy-cost trade-off:

$$\mathrm{KL}\left(\Pi\|\mathbb{Q}_{0,T}\right) = \mathbb{E}_\Pi\left[-\log q_{0,T}(X_0,X_T)\right] - \mathcal{H}(\Pi) + \text{const},$$

so minimizing KL is equivalent to minimizing a transport cost induced by the reference dynamics while maximizing the coupling entropy $\mathcal{H}(\Pi)$. For instance, when $\mathbb{Q}$ is a Brownian diffusion with constant noise scale, $-\log q_{0,T}(x,y)$ reduces to a quadratic cost, and the noise intensity controls the strength of the entropic regularization. As the noise decreases, the solution approaches classical OT, whereas larger noise yields higher-entropy, more mass-splitting couplings.

When the reference $\mathbb{Q}$ is Markov with transition density $p(s,\cdot\,;t,\cdot\,)$, assuming densities exist, the SB solution follows strictly positive potentials $\{\phi_t,\psi_t\}_{t\in[0,T]}$ such that

$$p^\star_t(x) = \phi_t(x)\psi_t(x)\,q_t(x), \tag{5}$$

and the optimal endpoint coupling can be written as

$$\Pi^\star_{0,T}(dx,dy) = \phi_0(x)\,p(0,x;T,y)\,\psi_T(y)\,dx\,dy, \tag{6}$$

where q_t denotes the marginal density of $\mathbb{Q}$ at time t, and (ϕ_0,ψ_T) are determined by the marginal constraints. It is often convenient to express a path measure through endpoint conditioning of the reference process. Let $\mathbb{Q}_{|0,T}(\cdot\mid x_0,x_T)$ denote the $\mathbb{Q}$-bridge, i.e., the conditional law of $X_{0:T}$ under $\mathbb{Q}$ given $(X_0,X_T)=(x_0,x_T)$. Then any *mixture* of $\mathbb{Q}$-bridges can be written as

$$\Pi(\cdot) = \int \mathbb{Q}_{|0,T}(\cdot\mid x_0,x_T)\,\Pi_{0,T}(dx_0,dx_T). \tag{7}$$

The endpoint-coupling view in Eq. (5)–(6) provides the key perspective used in our method. We will construct PDE-specific empirical endpoint couplings and learn bridge dynamics for forward and inverse reconstruction under these fixed couplings. For completeness, we defer the Markovian/reciprocal projection viewpoint of Schrödinger bridges and a brief review of diffusion Schrödinger bridge matching to Appendix A.

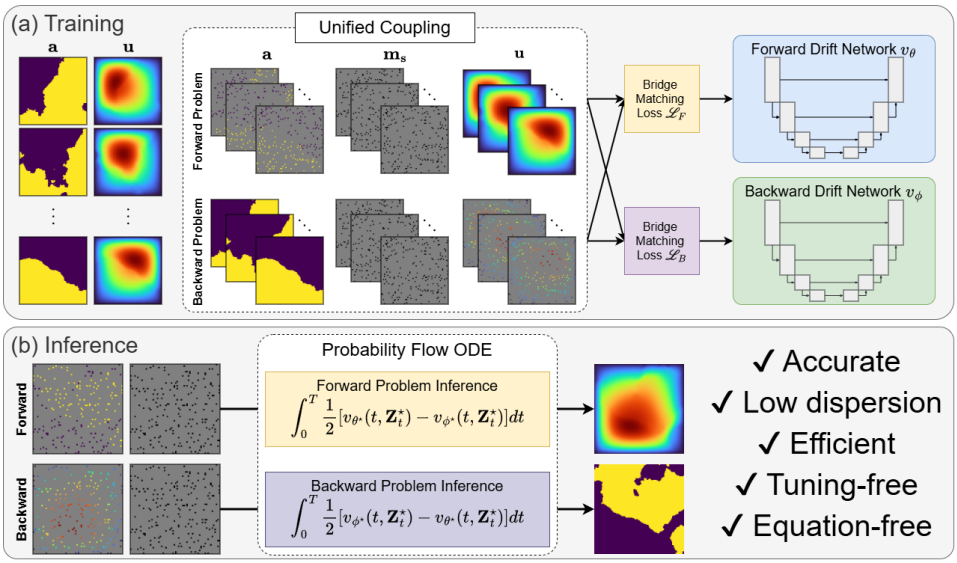


Figure 2: **Overview of SBNO. (a) Training.** From paired PDE samples $(\mathbf{a}, \mathbf{u})$ and a random observation mask $\mathbf{m_s}$, SBNO lifts each partially observed field by concatenating masked values and the binary mask. It constructs a unified endpoint coupling that mixes forward and inverse pairs while preserving empirical input-solution pairing. Forward/backward drift networks are trained with diffusion Schrödinger bridge matching losses. **(b) Inference.** Given partial observations and their mask, SBNO produces a deterministic reconstruction of the target field using the probability-flow ODE constructed from the learned drift pair, with the same trained model supporting both PDE tasks without observation or PDE guidance-weight tuning.

4 Method

In this section, we present a model for forward and inverse PDE reconstruction under partial observations using fixed-coupling diffusion bridge matching. We begin by defining the problem and then explain how the Schrödinger bridge framework applies in this context. Applied to PDE data, this approach learns bridge dynamics from endpoint couplings constructed using empirical input-solution pairings and partial-observation masks.

4.1 Problem Definition

Let $\mathcal{X} \subset \mathbb{R}^{d_x}$ be a spatial domain and let $\mathcal{T} \subset \mathbb{R}$ be a time domain. To cover both static and time-dependent PDEs under a unified notation, define the evaluation domain

$$\mathcal{Z} := \begin{cases} \mathcal{X}, & \text{(static PDEs, e.g., Darcy flow)} \\ \mathcal{T} \times \mathcal{X}, & \text{(time-dependent PDEs).} \end{cases} \tag{8}$$

Consider a family of PDEs parameterized by an input function $\mathbf{a} \in \mathcal{A}$, where $\mathbf{a}$ may encode coefficients, source terms, or initial/boundary conditions. The corresponding solution operator is denoted by $\mathbf{u} = \mathcal{G}(\mathbf{a})$ with $\mathcal{G} : \mathcal{A} \to \mathcal{U}$, where $\mathcal{U}$ is a function space over $\mathcal{Z}$. The solution field is denoted by $\mathbf{u}(z)$, where $\mathbf{u} : \mathcal{Z} \to \mathbb{R}^{d_u}$ may be scalar- or vector-valued (e.g., $\mathbf{u}(\cdot, \cdot)$ for the time-dependent case and $\mathbf{u}(\cdot)$ for the static case).

Let $\mathcal{S}$ denote the space of specifications, such as sensor locations, masks, or sampling patterns. For each $\mathbf{s} \in \mathcal{S}$, we denote the corresponding partial observation of the input function as $\mathbf{a_s}$, and the partial observation of the solution as $\mathbf{u_s}$. We assume a data distribution supported on paired samples $(\mathbf{a}, \mathbf{u})$ satisfying $\mathbf{u} = \mathcal{G}(\mathbf{a})$. Under specification $\mathbf{s} \in \mathcal{S}$, the ideal forward and inverse reconstruction targets can be viewed as conditional laws:

$$p_{\mathrm{F}}(\mathbf{u}|\mathbf{a_s}, \mathbf{s}) \quad \text{consistent with} \quad \mathbf{u} = \mathcal{G}(\mathbf{a}) \quad , \quad p_{\mathrm{I}}(\mathbf{a}|\mathbf{u_s}, \mathbf{s}) \quad \text{consistent with} \quad \mathbf{u} = \mathcal{G}(\mathbf{a}). \tag{9}$$

These distributions represent the probabilistic uncertainty over the unobserved regions of the solution $\mathbf{u}$ and the input parameters $\mathbf{a}$, respectively. In general, the inverse mapping from $\mathbf{u}$ to $\mathbf{a}$

is ill-posed and may be non-unique, especially under partial observations. Therefore, $p_{\mathrm{I}}(\mathbf{a}|\mathbf{u_s}, \mathbf{s})$ describes a potentially non-unique reconstruction target, for which we learn bridge dynamics and evaluate deterministic point reconstruction in the main experiments. We propose to learn bridge dynamics for these two reconstruction directions within a single unified fixed-coupling framework.

4.2 State Representation and Bridge Endpoints

To accommodate practical scenarios, we represent the fields in the function spaces $\mathcal{A}$ and $\mathcal{U}$ using discretized forms. After discretizing the evaluation domain $\mathcal{Z}$ with N grid points, each field is stored as a tensor (e.g., of shape (C, H, W) on a 2D grid) and flattened into a vector in $\mathbb{R}^D$ with total dimension $D := C{\cdot}N$. Given a specification $\mathbf{s} \in \mathcal{S}$ and a discretized field $\mathbf{x} \in \mathbb{R}^D$, let $\mathbf{m_s} \in \{0,1\}^D$ be the associated binary mask. We define the masked observation $\mathbf{x_s} := \mathbf{m_s} \odot \mathbf{x} \in \mathbb{R}^D$ and the corresponding *lifted* state representation:

$$\mathcal{E}_{\mathbf{s}}(\mathbf{x}) := \begin{bmatrix} \mathbf{m_s} \odot \mathbf{x} \\ \mathbf{m_s} \end{bmatrix} \in \mathbb{R}^{2D}. \tag{10}$$

For a fully observed field, the mask is taken as the all-ones vector $\mathbf{1}$, yielding $\mathcal{E}_{\mathbf{1}}(\mathbf{x}) = [\mathbf{x}; \mathbf{1}]$. Based on this representation, we construct the endpoint pairs for the forward and inverse problems. Let $(\mathbf{a}, \mathbf{u})$ denote an input-solution pair consistent with the PDE family, i.e., $\mathbf{u} = \mathcal{G}(\mathbf{a})$. We define two distinct endpoint couplings.

For the **forward operator**, given partial input $\mathbf{s} \in \mathcal{S}$, we define the source state $\mathbf{X}_0^{\mathrm{F}} := \mathcal{E}_{\mathbf{s}}(\mathbf{a})$ and the target state $\mathbf{X}_T^{\mathrm{F}} := \mathcal{E}_{\mathbf{1}}(\mathbf{u})$ (full solution). The induced endpoint coupling is the joint law:

$$\Pi_{0,T}^{\mathrm{F}} := \mathrm{Law}(\mathbf{X}_0^{\mathrm{F}}, \mathbf{X}_T^{\mathrm{F}}) \in \mathcal{P}(\mathbb{R}^{2D} \times \mathbb{R}^{2D}), \tag{11}$$

where $\mathrm{Law}(\cdot)$ denotes the distribution induced by the underlying PDE dataset. Conversely, for the **inverse operator**, we define the source state $\mathbf{X}_0^{\mathrm{I}} := \mathcal{E}_{\mathbf{1}}(\mathbf{a})$ (full input) and the target state $\mathbf{X}_T^{\mathrm{I}} := \mathcal{E}_{\mathbf{s}}(\mathbf{u})$ (partial solution), with the corresponding coupling:

$$\Pi_{0,T}^{\mathrm{I}} := \mathrm{Law}(\mathbf{X}_0^{\mathrm{I}}, \mathbf{X}_T^{\mathrm{I}}) \in \mathcal{P}(\mathbb{R}^{2D} \times \mathbb{R}^{2D}). \tag{12}$$

The conditional operators of interest are given by the endpoint conditionals derived from couplings:

$$p_F(\cdot|x_0) := \Pi_{0,T}^F(dx_T|\mathbf{X}_0^F = x_0) \quad , \quad p_I(\cdot|x_T) := \Pi_{0,T}^I(dx_0|\mathbf{X}_T^I = x_T). \tag{13}$$

These endpoint conditionals define the target reconstruction laws associated with the task-specific couplings. Their exact recovery by the learned Markovian dynamics or deterministic ODE is not guaranteed. Specifically, for the forward problems, sampling $\mathbf{X}_T \sim p_{\mathrm{F}}(\cdot|\mathbf{X}_0^{\mathrm{F}})$ predicts the full solution $\widehat{\mathbf{u}}$ from partial input observations. For the inverse problems, sampling $\mathbf{X}_0 \sim p_{\mathrm{I}}(\cdot|\mathbf{X}_T^{\mathrm{I}})$ recovers the full input $\widehat{\mathbf{a}}$ from partial solution observations. The lifted representation (10) places both partial observations and full fields in the shared space $\mathbb{R}^{2D}$, enabling unified learning of the drift fields.

4.3 Schrödinger-Bridge-Inspired Fixed-Coupling Formulations

This section formulates the forward and inverse reconstruction problems using a fixed-coupling diffusion bridge construction inspired by Schrödinger bridges. Rather than solving the classical Schrödinger bridge problem by optimizing over all endpoint couplings with prescribed marginals, we preserve the empirical endpoint couplings induced by paired PDE samples and learn Markovian drift fields associated with the resulting reciprocal bridge mixtures on the common state space $\mathbb{R}^{2D}$.

Let $\Omega := C([0,T], \mathbb{R}^{2D})$ be the path space of continuous trajectories $\mathbf{X}_{0:T} = (\mathbf{X}_t)_{t\in[0,T]}$. Fix a reference path measure $\mathbb{Q}$ on Ω induced by the SDE, $d\mathbf{X}_t = f_t(\mathbf{X}_t)\,dt + \sigma_t\,d\mathbf{B}_t$, where $\mathbf{B}_t$ is a standard Brownian motion in $\mathbb{R}^{2D}$, $f_t : \mathbb{R}^{2D} \to \mathbb{R}^{2D}$ is a drift, and $\sigma_t > 0$ is a scalar diffusion coefficient. The parameter $t \in [0,T]$ denotes the time, which is independent of the physical coordinates $z \in \mathcal{Z}$.

Classical Schrödinger bridges are specified by endpoint marginal constraints and implicitly determine an entropy-regularized optimal endpoint coupling. In contrast, our setting provides paired PDE samples, and therefore an empirical endpoint coupling $\Pi_{0,T}^{\alpha}$ is already available. We keep this empirical endpoint coupling fixed and define a reciprocal bridge mixture $\widetilde{P}^{\alpha} := \Pi_{0,T}^{\alpha} Q_{|0,T}$, where $Q_{|0,T}$

denotes the reference bridge conditioned on the endpoints. The learning problem is then to approximate the Markovian projection of this fixed-coupling reciprocal mixture by time-dependent drift fields. This fixed-coupling construction preserves the empirical input-solution pairing in the training data while providing bridge dynamics that can be used for forward and inverse reconstruction.

4.4 Unified Bridge Matching and Training Objective for Forward and Inverse Operators

To model both the forward and inverse problems within a single framework, we construct a fixed empirical endpoint coupling that mixes the two task-specific couplings while preserving the empirical input-solution pairing between a and u, i.e., $u = G(a)$. We then train a pair of time-dependent drifts on bridge samples drawn from this fixed-coupling mixture.

Two endpoint couplings $\Pi^{\mathrm{F}}_{0,T}$ and $\Pi^{\mathrm{I}}_{0,T}$ defined in Section 4.2 can be combined into a single coupling by mixing these two distributions $\Pi_{0,T} := \frac{1}{2}\,\Pi^{\mathrm{F}}_{0,T} + \frac{1}{2}\,\Pi^{\mathrm{I}}_{0,T}$. This mixture defines the fixed reciprocal bridge mixture used for training:

$$\widetilde{P}^{\Pi} := \Pi_{0,T} Q_{|0,T} = \int Q_{|0,T}(\cdot \mid x_0, x_T)\, \Pi_{0,T}(dx_0, dx_T). \tag{14}$$

Importantly, $\Pi_{0,T}$ is prescribed by the empirical paired-data construction and is not optimized over $\Pi(\pi_0, \pi_T)$. Let $M \in \{\mathrm{F}, \mathrm{I}\}$ be a binary indicator with $p(M = \mathrm{F}) = 1/2$. Sampling $(\mathbf{a}, \mathbf{u})$ from the paired PDE dataset and $\mathbf{s} \sim p(\mathbf{s})$, we can define the joint random variables as:

$$(\mathbf{X}_0, \mathbf{X}_T) = \begin{cases} (\mathcal{E}_{\mathbf{s}}(\mathbf{a}),\ \mathcal{E}_{\mathbf{1}}(\mathbf{u})), & M = \mathrm{F}, \\ (\mathcal{E}_{\mathbf{1}}(\mathbf{a}),\ \mathcal{E}_{\mathbf{s}}(\mathbf{u})), & M = \mathrm{I}, \end{cases} \tag{15}$$

where $\Pi_{0,T} = \mathrm{Law}(X_0, X_T)$. By construction, $\Pi_{0,T}$ is supported only on endpoint pairs generated from paired input-solution samples (a, u). The mixture changes the reconstruction direction, but not the underlying input-solution pairing.

Given an endpoint pair $(\mathbf{X}_0, \mathbf{X}_T) \sim \Pi_{0,T}$ and a bridge time $\tau \sim \mathrm{Unif}[0, T]$, an intermediate state is sampled from the τ-marginal of the bridge $\mathbf{X}_\tau \sim \mathbb{Q}_{|0,T}(\mathbf{X}_\tau \in \cdot | \mathbf{X}_0, \mathbf{X}_T)$. This construction yields tuples $(\mathbf{X}_0, \mathbf{X}_\tau, \mathbf{X}_T)$ drawn from the mixture-of-bridges path measure. As in DSBM, two time-dependent drift fields v_θ and v_ϕ on $\mathbb{R}^{2D}$ are introduced:

$$v_\theta : [0, T] \times \mathbb{R}^{2D} \to \mathbb{R}^{2D}, \quad v_\phi : [0, T] \times \mathbb{R}^{2D} \to \mathbb{R}^{2D}. \tag{16}$$

Here, v_θ approximates the forward drift and v_ϕ approximates the backward drift. The networks use the same architecture with separate parameters and are optimized independently. Following the bridge-matching formulation reviewed in Section A.2, we train forward/backward drifts by regression on $\mathbb{Q}$-bridge samples. Define the Q-bridge conditional-score targets

$$s^{T|t}(X_T; x) := \sigma_t^2 \nabla_x \log \mathbb{Q}_{T|t}(\mathbf{X}_T | \mathbf{X}_t) \quad , \quad s^{t|0}(x; X_0) := \sigma_t^2 \nabla_x \log \mathbb{Q}_{t|0}(\mathbf{X}_t | \mathbf{X}_0). \tag{17}$$

We minimize the following regression losses

$$\begin{aligned} \mathcal{L}_{\mathrm{F}}(\theta) &:= \mathbb{E}\Big[\big\| s^{T|\tau}(X_T; X_\tau) - v_\theta(\tau, \mathbf{X}_\tau) \big\|^2 / \sigma_\tau^2 \Big], \\ \mathcal{L}_{\mathrm{B}}(\phi) &:= \mathbb{E}\Big[\big\| s^{\tau|0}(X_\tau; X_0) - v_\phi(\tau, \mathbf{X}_\tau) \big\|^2 / \sigma_\tau^2 \Big], \end{aligned} \tag{18}$$

where the expectation is taken over $\tau \sim \mathrm{Unif}[0, T]$, $(X_0, X_T) \sim \Pi_{0,T}$, and $X_\tau \sim \mathbb{Q}_{|0,T}(\cdot \mid X_0, X_T)$. Because $\Pi_{0,T}$ mixes forward and inverse endpoint pairs, a single model (v_θ, v_ϕ) is trained to support both partial-to-full directions through the shared lifted state space (10). Both drifts are required for our probability-flow ODE inference, including when training uses only one PDE-task coupling. We therefore use Schrödinger bridge matching as a bridge-dynamics learning tool under a fixed empirical endpoint coupling, rather than as a procedure for estimating the optimal coupling between endpoint marginals.

The learned bridge induces two conditional generative processes on $\mathbb{R}^{2D}$. For the forward problems, given a partial input $\mathbf{X}_0 = \mathcal{E}_{\mathbf{s}}(\mathbf{a})$, $\mathbf{X}_T$ can be sampled using the forward SDE with drift v_θ. The decoded value of $\mathbf{X}_T$ yields the full solution prediction $\widehat{\mathbf{u}}$. For the inverse problems, given a partial solution $\mathbf{X}_T = \mathcal{E}_{\mathbf{s}}(\mathbf{u})$, $\mathbf{X}_0$ can be sampled using the backward SDE with drift v_ϕ. The decoded value of $\mathbf{X}_0$ yields the full input prediction $\widehat{\mathbf{a}}$. The detailed training procedure can be found in Appendix B.

4.5 Deterministic Inference via Probability-flow ODE

Although inference can be performed stochastically by sampling trajectories from the learned SDEs, we adopt a deterministic approach to obtain a stable representative reconstruction. Specifically, we solve the associated *probability flow ODEs* of the learned bridge. For the fixed-coupling bridge mixture $\widetilde{P}^{\Pi}$, let $P^{\Pi} := \mathrm{proj}_{\mathcal{M}}(\widetilde{P}^{\Pi})$ denote its Markovian projection, with one-time marginal density p_t^{Π}. The corresponding forward and backward Markovian projection drifts, denoted by v_F^{Π} and v_B^{Π}, satisfy the standard forward-backward drift relation:

$$v_B^{\Pi}(t,x) = -v_F^{\Pi}(t,x) + \sigma_t^2 \nabla \log p_t^{\Pi}(x). \tag{19}$$

Under the standard regularity assumptions and the corresponding marginal initialization, this identity yields a probability-flow ODE with the same one-time marginals as the Markovian projection P^{Π} Concretely, the probability flow $\mathbf{Z}_t^{\Pi}$ has the same one-time marginals as $\mathbb{P}^{\Pi}$, i.e., $\mathrm{Law}(\mathbf{Z}_t^{\Pi}) = \mathbb{P}_t^{\Pi}$ for all $t \in [0,T]$, and is governed by:

$$dZ_t^{\Pi} = \left\{ f_t(Z_t^{\Pi}) + \frac{1}{2}\left[v_F^{\Pi}(t, Z_t^{\Pi}) - v_B^{\Pi}(t, Z_t^{\Pi}) \right] \right\} dt. \tag{20}$$

In practice, (v_F^{Π} ,v_B^{Π}) are replaced by the learned networks (v_θ, v_ϕ). For forward inference we initialize $\mathbf{Z}_0 = \mathbf{X}_0^{\mathrm{F}} = \mathcal{E}_{\mathbf{s}}(\mathbf{a})$ and for inverse inference we initialize $\mathbf{Z}_T = \mathbf{X}_T^{\mathrm{I}} = \mathcal{E}_{\mathbf{s}}(\mathbf{u})$. Note that this probability-flow ODE matches only the one-time marginals $\mathbb{P}_t^{\Pi}$ of the Markovian projected fixed-coupling bridge. It does not reproduce the full stochastic path law of the bridge mixture, nor does it preserve the empirical endpoint coupling $\Pi_{0,T}$. Consequently, the deterministic endpoint relation induced by the ODE generally differs from the fixed empirical coupling used during bridge matching. For a fixed observation, its endpoint is a deterministic point reconstruction, not necessarily the conditional mean, and it cannot represent multimodal uncertainty. However, for PDE solving where the goal is a representative reconstruction rather than sampling diverse solutions, this deterministic flow provides a practical single prediction following the learned marginal evolution.

5 Experiments

In this section, we present experiments designed to validate the proposed method in terms of PDE reconstruction accuracy, error dispersion, and inference efficiency. They are conducted under noiseless observations, and SBNO inference is performed deterministically throughout. This setting allows for a clearer comparison of the three aspects above under a controlled evaluation protocol. To further examine broader use cases, we provide additional experiments on noisy partial observations and stochastic inference for predictive-uncertainty diagnostics in Appendix J and Appendix I, respectively.

5.1 Experimental Setup

Datasets We evaluate both forward and inverse problem-solving tasks under partial observations on five PDE benchmarks: four datasets from the DiffusionPDE [10] and one dataset from The Well [27]. All datasets are discretized on a (128×128) spatial grid. During inference, we utilize only 5% of the available observations, corresponding to 819 points. Details about the datasets can be found in Appendix D. Beyond the five main benchmarks, we further evaluate SBNO on three smaller-scale PDE datasets with distinct dynamical characteristics, as reported in Appendix H.

Table 1: Qualitative comparison of SBNO and the baseline configurations evaluated in this study.

Capability	SBNO (ours)	FunDPS & DiffusionPDE	FNOPE	GNOT	FNO
Observation conditioning	mask-aware	only observation	only observation	only observation	mask-aware
Unified PDE-task training	**yes**	**yes**	no	no	no
Inference-time hyperparameters	**not required**	required	**not required**	**not required**	**not required**
PDE information	**equation-free**	PDE guidance	**equation-free**	**equation-free**	**equation-free**
Inference type	**both possible**	stochastic	stochastic	deterministic	deterministic

Baseline Methods We compare SBNO against representative baselines from three categories: guided diffusion-based PDE solvers, deterministic neural operators, and probabilistic neural operators. Specifically, we include DiffusionPDE [10] and FunDPS [38] as diffusion-based baselines,

Table 2: The evaluation results are reported on five benchmark datasets. Each performance metric is expressed as the "mean ± std" of the relative L_2 error across test samples. For clarity, the best values in each column are highlighted in **bold**, while the second-best values are marked with underline.

Dataset	Problem	SBNO (Ours)	FunDPS	DiffusionPDE	FNOPE	GNOT	FNO
Darcy flow	Forward	**0.0359 ± 0.0133**	0.0461 ± 0.0207	0.0552 ± 0.0849	0.2219 ± 0.0893	0.0510 ± 0.0148	0.0582 ± 0.0135
	Inverse	**0.0269 ± 0.0125**	0.0835 ± 0.0380	0.0499 ± 0.0338	0.2241 ± 0.1110	0.0779 ± 0.0367	0.0946 ± 0.0409
Poisson	Forward	**0.0424 ± 0.0219**	0.0775 ± 0.0473	0.0986 ± 0.1696	0.7849 ± 0.4061	0.0929 ± 0.0428	0.1458 ± 0.0518
	Inverse	0.2167 ± **0.0345**	0.3664 ± 0.1232	**0.2015** ± 0.0792	0.8423 ± 0.2244	0.3585 ± 0.0910	0.3912 ± 0.0991
Helmholtz	Forward	**0.0519** ± 0.0315	0.0563 ± **0.0310**	0.0902 ± 0.1412	0.7388 ± 0.3684	0.0926 ± 0.0393	0.2191 ± 0.0854
	Inverse	0.2310 ± **0.0501**	0.2550 ± 0.0740	**0.1893** ± 0.2557	0.7111 ± 0.2000	0.3728 ± 0.0865	0.4178 ± 0.0947
Nonbounded Navier-Stokes	Forward	**0.0368 ± 0.0086**	0.0435 ± 0.0136	0.0922 ± 0.2189	1.1290 ± 0.2625	0.2360 ± 0.0476	0.2470 ± 0.0437
	Inverse	**0.1097 ± 0.0167**	0.1241 ± 0.0234	0.1584 ± 0.2362	1.8253 ± 0.4081	0.3669 ± 0.0629	0.4106 ± 0.0557
Gray-Scott	Forward	**0.0393** ± 0.0193	0.0846 ± 0.0222	0.0468 ± 0.0545	0.0930 ± **0.0109**	0.1025 ± 0.0084	0.0857 ± 0.0112
	Inverse	**0.0382** ± 0.0167	0.0683 ± 0.0225	0.0589 ± 0.0622	0.1094 ± **0.0089**	0.0871 ± 0.0084	0.0724 ± 0.0102

FNO [17] and GNOT [8] as deterministic neural-operator baselines, and FNOPE [23] as a probabilistic operator-learning baseline. For methods that do not natively share a unified forward/inverse objective, we train separate models for each task when applicable. Further details on the baseline input representations, inference protocols, and implementation choices are provided in Appendix E. We summarize the qualitative comparison between SBNO and the baselines in Table 1. The FunDPS results are obtained from our reproduced configurations; we did not recover its originally reported accuracy, and Appendix E documents the protocol and unresolved reproduction discrepancy.

Training and Inference Protocol We parameterize the forward/backward drifts of SBNO, v_θ and v_ϕ, using a time-conditioned U-Net. At inference time, SBNO produces reconstructions deterministically by integrating the probability-flow ODE induced by the learned Schrödinger bridge. We solve this ODE using a fourth-order Runge-Kutta (RK4) solver with 32 integration steps. Accordingly, during training, we compute the bridge-matching objective by sampling only the discrete time points that coincide with the 32-step RK4 trajectory, and we resample sensor locations at every iteration while keeping the observation ratio fixed at 5% in the main experiments. For diffusion-based baselines, we tune inference-time hyperparameters on the validation set. A hyperparameter sensitivity analysis is provided in Appendix F, and the exact configurations for each model are reported in Appendix O.3.

5.2 Results

Prediction Error We evaluate reconstruction accuracy using the relative L_2 error and report both the mean and the test-instance error dispersion, namely "std," as described in Appendix C. Table 2 summarizes results on the benchmark datasets under 5% partial observations. Overall, SBNO achieves the best or second-best mean relative L_2 error in all ten forward and inverse benchmark settings. Moreover, SBNO attains relatively low dispersion among competitive baselines with comparable mean performance, reducing the average error dispersion by 44.91%, from 0.0416 for FunDPS to 0.0229. These results suggest that SBNO combines strong average accuracy with lower test-instance error dispersion in the sparse-observation regime. For all the considered baselines, SBNO remains competitive against diffusion-based methods while also outperforming the deterministic and probabilistic operator baselines in most settings.

Inference Efficiency We next evaluate the inference-time efficiency of SBNO under the same partial-observation setting. This comparison is restricted to the diffusion-based baselines, FunDPS and DiffusionPDE, as they are the most relevant generative baselines for comparison with SBNO. We evaluate both metrics over the test set, as described in Appendix C. All results are averaged over five runs. Table 3 summarizes the findings. Overall, SBNO demonstrates substantially lower inference cost than the guided diffusion baselines. It is at least $9.7\times$ faster in wall-clock time per sample and consumes at least $4.3\times$ less peak GPU memory. Combined with its competitive reconstruction accuracy, these measurements

Table 3: Cost comparison between diffusion-based models. We report wall-clock time per sample and peak GPU memory. The best values are highlighted in **bold**.

Method	Time per Sample (sec)	Peak Mem (MB)
DiffusionPDE	46.65	3,846.7
FunDPS	2.71	5,428.8
SBNO (ours)	**0.28**	**886.0**

indicate a favorable accuracy–cost trade-off relative to the reproduced guided-diffusion baselines in the evaluated setting. We additionally analyze the trade-off between reconstruction performance and inference cost by varying the ODE solver steps, with results reported in Appendix K.

6 Conclusion

In this work, we address the problem of PDE forward and inverse operator learning under sparse observations. We introduce SBNO, a Schrödinger-bridge-inspired neural operator that uses a mask-aware PDE endpoint coupling and diffusion bridge matching to support forward and inverse reconstruction with one trained drift pair. It performs deterministic point reconstruction through the probability-flow ODE constructed from the learned drifts, without observation or PDE guidance-weight tuning. Across five PDE benchmarks with 5% point-wise observations, SBNO achieves competitive or superior reconstruction accuracy while reducing test-instance error dispersion. Moreover, SBNO significantly reduces inference cost compared to guided diffusion solvers, being at least $9.7\times$ faster and using at least $4.3\times$ less peak GPU memory, making it a practical and scalable alternative for repeated reconstructions under sparse observations.

Limitations and Future Works While the results are strong in the current benchmark setting, the present implementation of SBNO still assumes discretized fields defined on a grid. As a result, although the model can handle partial observations on that grid through masking, it cannot directly ingest off-grid observations at arbitrary continuous sensor coordinates without first projecting or interpolating them onto the underlying grid. For fixed observations, deterministic ODE inference returns a single point reconstruction and does not guarantee recovery of the conditional mean or preservation of the empirical endpoint coupling, nor can it represent multimodal uncertainty. This limitation stems from the current grid-based backbone rather than from the Schrödinger bridge formulation itself. Extending SBNO to architectures that can natively process irregular sensor observations is an important direction for future work.

7 Acknowledgements

This work was supported by Samsung Electronics Co., Ltd(IO250616-13070-01, 50%), Samsung Research Funding & Incubation Center for Future Technology of Samsung Electronics under Project Number SRFC-IT2402-08 (49%) and the Institute for Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korean government (MSIT) (No. RS-2025-25442149, LG AI STAR Talent Development Program for Leading Large-Scale Generative AI Models in the Physical AI Domain, 1%).

References

[1] Chen-Fu Chien, Chia-Yu Hsu, and Chih-Wei Hsiao. Manufacturing intelligence to forecast and reduce semiconductor cycle time. *Journal of Intelligent Manufacturing*, 23:2281–2294, 2012.

[2] Sung Woong Cho and Hwijae Son. Physics-informed deep inverse operator networks for solving pde inverse problems. In *The Thirteenth International Conference on Learning Representations*, 2025.

[3] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in neural information processing systems*, 34:17695–17709, 2021.

[4] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[5] Hans Föllmer. Random fields and diffusion processes. *Lecture Notes in Mathematics*, pages 101–203, 1988.

[6] Promit Ghosal, Marcel Nutz, and Espen Bernton. Stability of entropic optimal transport and schrödinger bridges. *Journal of Functional Analysis*, 283(9):109622, 2022.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[8] Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. Gnot: A general neural operator transformer for operator learning. In *International conference on machine learning*, pages 12556–12569. PMLR, 2023.

[9] Bastian Harrach. An introduction to finite element methods for inverse coefficient problems in elliptic pdes. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 123(3):183–210, 2021.

[10] Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. Diffusionpde: Generative pde-solving under partial observation. *Advances in Neural Information Processing Systems*, 37:130291–130323, 2024.

[11] Victor Isakov. *Inverse problems for partial differential equations*. Springer, 2006.

[12] Ameya D Jagtap, Zhiping Mao, Nikolaus Adams, and George Em Karniadakis. Physics-informed neural networks for inverse problems in supersonic flows. *Journal of Computational Physics*, 466:111402, 2022.

[13] Felix Koehler, Simon Niedermayr, Rüdiger Westermann, and Nils Thuerey. Apebench: A benchmark for autoregressive neural emulators of pdes. *Advances in Neural Information Processing Systems*, 37:120252–120310, 2024.

[14] Christian Léonard. Girsanov theory under a finite entropy condition. In *Séminaire de Probabilités XLIV*, pages 429–465. Springer, 2012.

[15] Christian Léonard. A survey of the schrödinger problem and some of its connections with optimal transport. *Discrete and Continuous Dynamical Systems*, 34(4):1533–1574, 2013.

[16] Christian Léonard, Sylvie Rœlly, and Jean-Claude Zambrini. Reciprocal processes. a measure-theoretical point of view. *Probability Surveys*, 11:237–269, 2014.

[17] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, Anima Anandkumar, et al. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

[18] Jae Hyun Lim, Nikola B Kovachki, Ricardo Baptista, Christopher Beckham, Kamyar Azizzadenesheli, Jean Kossaifi, Vikram Voleti, Jiaming Song, Karsten Kreis, Jan Kautz, et al. Score-based diffusion models in function space. *Journal of Machine Learning Research*, 26 (158):1–62, 2025.

[19] Levi E Lingsch, Dana Grund, Siddhartha Mishra, and Georgios Kissas. Fuse: Fast unified simulation and estimation for pdes. *Advances in Neural Information Processing Systems*, 37: 21656–21698, 2024.

[20] Da Long, Zhitong Xu, Qiwei Yuan, Yin Yang, and Shandian Zhe. Invertible fourier neural operators for tackling both forward and inverse problems. In *International Conference on Artificial Intelligence and Statistics*, pages 3043–3051. PMLR, 2025.

[21] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

[22] Roberto Molinaro, Yunan Yang, Björn Engquist, and Siddhartha Mishra. Neural inverse operators for solving pde inverse problems. In *International Conference on Machine Learning*, pages 25105–25139. PMLR, 2023.

[23] Guy Moss, Leah Sophie Muhle, Reinhard Drews, Jakob H Macke, and Cornelius Schröder. Fnope: Simulation-based inference on function spaces with fourier neural operators. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[24] Stephen C Myers and Craig A Schultz. Improving sparse network seismic location with bayesian kriging and teleseismically constrained calibration events. *Bulletin of the Seismological Society of America*, 90(1):199–211, 2000.

[25] Nicholas H. Nelsen and Yunan Yang. Operator learning meets inverse problems: A probabilistic perspective. *arXiv preprint arXiv:2508.20207*, 2025.

[26] Zachary G Nicolaou, Guanyu Huo, Yihui Chen, Steven L Brunton, and J Nathan Kutz. Data-driven discovery and extrapolation of parameterized pattern-forming dynamics. *Physical Review Research*, 5(4):L042017, 2023.

[27] Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina Agocs, Miguel Beneitez, Marsha Berger, Blakesly Burkhart, Stuart Dalziel, Drummond Fielding, et al. The well: a large-scale collection of diverse physics simulations for machine learning. *Advances in Neural Information Processing Systems*, 37:44989–45037, 2024.

[28] Alfio Quarteroni and Alberto Valli. *Numerical approximation of partial differential equations*, volume 23. Springer Science & Business Media, Berlin, 2008.

[29] Md Ashiqur Rahman, Manuel A Florez, Anima Anandkumar, Zachary E Ross, and Kamyar Azizzadenesheli. Generative adversarial neural operators. *Transactions on Machine Learning Research*, 2022.

[30] 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*, 378:686–707, 2019.

[31] Nusrat Rouf, Majid Bashir Malik, Tasleem Arif, Sparsh Sharma, Saurabh Singh, Satyabrata Aich, and Hee-Cheol Kim. Stock market prediction using machine learning techniques: a decade survey on methodologies, recent developments, and future directions. *Electronics*, 10 (21):2717, 2021.

[32] Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 36:62183–62223, 2023.

[33] Andrew M Stuart. Inverse problems: a bayesian perspective. *Acta numerica*, 19:451–559, 2010.

[34] Shashank Subramanian, Peter Harrington, Kurt Keutzer, Wahid Bhimji, Dmitriy Morozov, Michael W. Mahoney, and Amir Gholami. Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[35] Albert Tarantola. *Inverse problem theory and methods for model parameter estimation*. SIAM, 2005.

[36] Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. Integrating scientific knowledge with machine learning for engineering and environmental systems. *ACM Computing Surveys*, 55(4):1–37, 2022.

[37] Liu Yang, Xuhui Meng, and George Em Karniadakis. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.

[38] 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. *arXiv preprint arXiv:2505.17004*, 2025.

[39] Jeremy Yu, Lu Lu, Xuhui Meng, and George Em Karniadakis. Gradient-enhanced physics-informed neural networks for forward and inverse pde problems. *Computer Methods in Applied Mechanics and Engineering*, 393:114823, 2022.

[40] Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. *arXiv preprint arXiv:2309.16948*, 2023.

A Additional Theoretical Background

A.1 Markovian and Reciprocal Projections

The Schrödinger bridge $\mathbb{P}^\star$ can be characterized through two structural constraints: (i) Markovianity and (ii) preservation of the $\mathbb{Q}$-bridge. "Preserving the Q-bridge" means that the conditional path law given endpoints $P(\cdot \mid X_0 = x_0, X_T = x_T) = Q_{|0,T}(\cdot \mid x_0, x_T)$ is fixed for all (x_0, x_T), while only the endpoint coupling may vary. These are formalized via two projection operators acting on path measures. Under mild assumptions, the $\mathbb{Q}$-bridge is itself a diffusion bridge and satisfies the Doob h-transform representation

$$\begin{aligned} d\mathbf{X}_t^{0,T} =& \Big\{ f_t(\mathbf{X}_t^{0,T}) + \sigma_t^2 \nabla \log \mathbb{Q}_{T|t}(x_T|\mathbf{X}_t^{0,T}) \Big\} dt \\ &+ \sigma_t \, d\mathbf{B}_t, \quad \mathbf{X}_0^{0,T} = x_0, \end{aligned} \tag{21}$$

where $\mathbb{Q}_{T|t}(\cdot|x)$ denotes the conditional density of $\mathbf{X}_T$ given $\mathbf{X}_t = x$ under $\mathbb{Q}$ and ∇ denotes the gradient with respect to the state variable X_t.

Markovian Projection Consider a path measure $\Pi \in \mathcal{P}(\Omega)$ (typically a mixture of bridges). Let $\mathcal{M}$ be the class of Markov diffusion processes with diffusion coefficient σ_t. The *Markovian projection* of Π, denoted $\mathbb{M}^\star = \mathrm{proj}_{\mathcal{M}}(\Pi)$, is the unique element in $\mathcal{M}$ under standard conditions. It is associated with the SDE:

$$d\mathbf{X}_t^\star = \big\{ f_t(\mathbf{X}_t^\star) + v_t^\star(\mathbf{X}_t^\star) \big\} dt + \sigma_t d\mathbf{B}t, \tag{22}$$

where the drift adjustment $v_t^\star$ is given by the conditional expectation over the target mixture Π:

$$v_t^\star(x) = \sigma_t^2 \mathbb{E}_\Pi \left[\nabla \log \mathbb{Q}_{T|t}(\mathbf{X}_T|x) \big| \mathbf{X}_t = x \right]. \tag{23}$$

Crucially, $\mathbb{M}^\star$ minimizes the reverse KL divergence from the mixture Π to the Markov manifold $\mathcal{M}$ [32]:

$$\mathbb{M}^\star = \arg\min_{\mathbb{M} \in \mathcal{M}} \mathrm{KL}(\Pi \| \mathbb{M}). \tag{24}$$

This projection effectively "marginalizes" the non-Markovian history dependencies into the current state via the drift term.

Reciprocal Projection The *reciprocal class* $\mathcal{R}(\mathbb{Q})$ consists of all measures that share the same bridge dynamics as $\mathbb{Q}$ but may have arbitrary endpoint couplings [16]. Formally,

$$\mathcal{R}(\mathbb{Q}) := \big\{ \Pi \in \mathcal{P}(\Omega) : \exists \Pi_{0,T} \text{ s.t. } \Pi = \Pi_{0,T} \mathbb{Q}_{|0,T} \big\}. \tag{25}$$

The *reciprocal projection* of a path measure $\mathbb{P}$ onto $\mathcal{R}(\mathbb{Q})$ simply replaces the bridge of $\mathbb{P}$ with that of $\mathbb{Q}$ while preserving the endpoint joint distribution [32]:

$$\mathrm{proj}_{\mathcal{R}(\mathbb{Q})}(\mathbb{P}) := \mathbb{P}_{0,T} \mathbb{Q}_{|0,T}. \tag{26}$$

SB as an Intersection The reciprocal projection ensures the correct bridge dynamics but may break Markovianity. Conversely, the Markovian projection ensures Markovianity but may distort the bridge. Among path measures satisfying $P_0 = \pi_0$ and $P_T = \pi_T$, the Schrödinger bridge is the unique element in $\mathcal{M} \cap \mathcal{R}(Q)$.

A.2 Diffusion Schrödinger Bridge Matching

Shi et al. [32] proposed *Iterative Markovian Fitting* (IMF) to compute the Schrödinger bridge by alternating projections between the Markov manifold $\mathcal{M}$ and the reciprocal class $\mathcal{R}(Q)$. Starting from an initial reciprocal path measure $P^{(0)} \in \mathcal{R}(Q)$ that satisfies the endpoint constraints $\mathbb{P}_0^{(0)} = \pi_0$ and $\mathbb{P}_T^{(0)} = \pi_T$, IMF defines the sequence $\{\mathbb{P}^{(n)}\}_{n \in \mathbb{N}}$ by

$$\mathbb{P}^{(2n+1)} = \mathrm{proj}_{\mathcal{M}}\big(\mathbb{P}^{(2n)}\big), \; \mathbb{P}^{(2n+2)} = \mathrm{proj}_{\mathcal{R}(Q)}\big(\mathbb{P}^{(2n+1)}\big), \tag{27}$$

where both projections preserve the endpoint marginals for all n. Under suitable conditions, the fixed point of this alternating-projection scheme coincides with the Schrödinger bridge $P^\star$. Since

Markovian projection $\mathrm{proj}_{\mathcal{M}}$ is generally intractable in high dimensions, DSBM [32] provides a scalable implementation by replacing the exact Markovian projection with regression-based *bridge matching*. Concretely, given a Markov iterate $\mathbb{M}^{(n)} \in \mathcal{M}$, the reciprocal update forms a reciprocal measure by preserving the current endpoint coupling:

$$\widetilde{\mathbb{P}}^{(n)} := \mathrm{proj}_{\mathcal{R}(\mathbb{Q})}\big(\mathbb{M}^{(n)}\big) = \mathbb{M}_{0,T}^{(n)}\mathbb{Q}_{|0,T}, \tag{28}$$

i.e., a mixture of Q-bridges with endpoint coupling $\mathbb{M}_{0,T}^{(n)} = \mathrm{Law}_{\mathbb{M}^{(n)}}(X_0, X_T)$. DSBM then updates the forward/backward drift fields by regressing to conditional-score targets of the reference Q-bridge evaluated on bridge samples, producing the next Markov iterate $M^{(n+1)} \approx \mathrm{proj}_{\mathcal{M}}(\widetilde{P}^{(n)})$. The updated endpoint coupling $\mathbb{M}_{0,T}^{(n+1)}$ is subsequently reused in Eq. (28) for the next reciprocal update.

We defer the explicit regression losses and the sampling procedure to Section 4.4, where we instantiate them with our PDE-specific endpoint constructions and the unified coupling.

B Training Algorithm

Algorithm 1 Unified DSBM bridge matching training for forward and inverse operators

Require: Paired PDE samples $(\mathbf{a}, \mathbf{u})$ with $\mathbf{u} = \mathcal{G}(\mathbf{a})$, specification distribution $p(\mathbf{s})$ and lifted map $\mathcal{E}_{\mathbf{s}}(\cdot)$ in Eq. (10), reference diffusion $\mathbb{Q}$ with bridge sampler $\mathbb{Q}_{|0,T}(\cdot|\mathbf{X}_0, \mathbf{X}_T)$ and conditional score oracles $\nabla \log \mathbb{Q}_{T|t}(\mathbf{X}_T|\mathbf{X}_t)$, $\nabla \log \mathbb{Q}_{t|0}(\mathbf{X}_t|\mathbf{X}_0)$, diffusion schedule σ_t, time horizon T, batch size B, iteration number N, time-conditioned U-Net drift networks $v_\theta(t, \mathbf{x})$ and $v_\phi(t, \mathbf{x})$.

for $n = 1, \dots, N$ **do**
  Sample a batch $\{(\mathbf{a}^{(i)}, \mathbf{u}^{(i)})\}_{i=1}^B$, specifications $\{\mathbf{s}^{(i)}\}_{i=1}^B \sim p(\mathbf{s})$, and mode indicators $M^{(n)}$ with $p(M^{(n)} = \mathrm{F}) = 1/2$.
  if $M^{(i)} = \mathrm{F}$ **then**
    $\mathbf{X}_0^{(i)}, \mathbf{X}_T^{(i)} \leftarrow \mathcal{E}_{\mathbf{s}^{(i)}}(\mathbf{a}^{(i)}), \mathcal{E}_{\mathbf{1}}(\mathbf{u}^{(i)})$ {Forward: partial input → full solution}
  else
    $\mathbf{X}_0^{(i)}, \mathbf{X}_T^{(i)} \leftarrow \mathcal{E}_{\mathbf{1}}(\mathbf{a}^{(i)}), \mathcal{E}_{\mathbf{s}^{(i)}}(\mathbf{u}^{(i)})$ {Inverse: partial solution → full input}
  end if
  Sample $\tau^{(i)} \sim \mathrm{Unif}[0, T]$, intermediate state $\mathbf{X}_{\tau^{(i)}}^{(i)} \sim \mathbb{Q}_{|0,T}(\mathbf{X}_{\tau^{(i)}}^{(i)} \in \cdot|\mathbf{X}_0^{(i)}, \mathbf{X}_T^{(i)})$.
  Compute minibatch losses (Monte Carlo estimates of Eq. (18))

$$\widehat{\mathcal{L}}_{\mathrm{F}}(\theta) \leftarrow \tfrac{1}{B}\sum_{i=1}^B \tfrac{1}{\sigma^2_{\tau^{(i)}}} \Big\| \sigma^2_{\tau^{(i)}} \nabla \log \mathbb{Q}_{T|\tau^{(i)}}\Big(\mathbf{X}_T^{(i)}|\mathbf{X}_{\tau^{(i)}}^{(i)}\Big) - v_\theta\big(\tau^{(i)}, \mathbf{X}_{\tau^{(i)}}^{(i)}\big)\Big\|^2$$

$$\widehat{\mathcal{L}}_{\mathrm{B}}(\phi) \leftarrow \tfrac{1}{B}\sum_{i=1}^B \tfrac{1}{\sigma^2_{\tau^{(i)}}} \Big\| \sigma^2_{\tau^{(i)}} \nabla \log \mathbb{Q}_{\tau^{(i)}|0}\Big(\mathbf{X}_{\tau^{(i)}}^{(i)}|\mathbf{X}_0^{(i)}\Big) - v_\phi\big(\tau^{(i)}, \mathbf{X}_{\tau^{(i)}}^{(i)}\big)\Big\|^2.$$

  Total loss: $\widehat{\mathcal{L}}(\theta, \phi) \leftarrow \widehat{\mathcal{L}}_{\mathrm{F}}(\theta) + \widehat{\mathcal{L}}_{\mathrm{B}}(\phi)$.
  Update parameters (θ, ϕ) by minimizing $\widehat{\mathcal{L}}(\theta, \phi)$.
end for
Return $v_\theta(t, \mathbf{x})$ and $v_\phi(t, \mathbf{x})$

B.1 Analytic Bridge Sampling and Regression Targets

Analytic bridge sampling and regression targets. The conditional-score targets in Algorithm 1 do not require separately trained score networks. For a zero-drift Gaussian reference

$$dX_t = \sigma_t \, dB_t,$$

with a deterministic scalar diffusion schedule, define the accumulated variance

$$V(t) := \int_0^t \sigma_s^2 \, ds.$$

Its transition laws are

$$Q_{t|0}(\cdot|X_0) = \mathcal{N}(X_0, V(t)I), \qquad Q_{T|t}(\cdot|X_t) = \mathcal{N}(X_t, [V(T) - V(t)]I).$$

For $0 < t < T$, a state conditioned on the endpoint pair (X_0, X_T) can therefore be sampled as

$$X_t = \left(1 - \frac{V(t)}{V(T)}\right) X_0 + \frac{V(t)}{V(T)} X_T + \sqrt{\frac{V(t)[V(T) - V(t)]}{V(T)}}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I).$$

The scaled conditional-score targets in Eq. (17) are

$$s_{T|t}(X_T; X_t) = \sigma_t^2 \frac{X_T - X_t}{V(T) - V(t)}, \qquad s_{t|0}(X_t; X_0) = \sigma_t^2 \frac{X_0 - X_t}{V(t)}.$$

For constant diffusion $\sigma_t = \sigma$ and $T = 1$, these expressions reduce to

$$X_t = (1-t)X_0 + tX_T + \sigma\sqrt{t(1-t)}\,\epsilon, \qquad s_{T|t}(X_T; X_t) = \frac{X_T - X_t}{1-t}, \qquad s_{t|0}(X_t; X_0) = \frac{X_0 - X_t}{t}.$$

The drift networks directly regress these analytic targets using the weighted objectives in Eq. (18). A small time cutoff is used to avoid evaluating the singular targets at the endpoints. Thus, "score oracle" denotes an analytically computable regression target, rather than an auxiliary learned model.

C Evaluation

We evaluate model performance using the relative L_2 error between predicted and ground-truth solution fields on the test set. For each i-th test sample , given the true solution y_i and the model prediction $\hat{y}_i$, the relative error is computed as:

$$e_i = \frac{|y_i - \hat{y}_i|_p}{|y_i|_p} \tag{29}$$

where $|\cdot|_p$ denotes the L_p norm (typically $p = 2$ in our experiments). The mean and standard deviation of the relative errors over all test samples are then reported to summarize overall model accuracy and variability.

$$\text{dataset error mean} = \frac{1}{N}\sum_{i=1}^{N} e_i \quad , \quad \text{dataset error std} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(e_i - \frac{1}{N}\sum_{i=1}^{N} e_i\right)^2} \tag{30}$$

We also report two efficiency metrics during inference: wall-clock time per sample and peak GPU memory usage. Wall-clock time is measured on a single NVIDIA B200 GPU using the largest feasible batch size for each method. To isolate pure inference cost, we exclude data loading. Peak memory is measured as the maximum GPU memory allocated during inference with a batch size of one. All results are averaged over five runs. Because time per sample is measured using method-specific batch sizes, it reflects batched inference throughput and should not be interpreted as batch-one request latency. The memory measurements use a separate batch-one protocol.

D Datasets

The main experiments in Section 5 consist of five PDE datasets: four datasets from the DiffusionPDE benchmark release [10] and one Gray-Scott reaction-diffusion dataset from The Well [27].[1] The four DiffusionPDE datasets include three static PDEs and one time-dependent Navier-Stokes dataset. For these four datasets, all fields are defined on the unit square domain $\Omega = (0, 1)^2$ and discretized on a 128×128 grid. Each dataset contains 50,000 training samples, and we use fixed validation and test sets of 100 and 900 samples, respectively.

The Gray-Scott dataset differs from the DiffusionPDE datasets in both its source and construction. It is obtained from The Well and consists of two-channel reaction-diffusion trajectories. Following the construction described in Appendix D.2.2, we form one-step forecasting pairs from stored trajectory frames, yielding 10,000 training samples and 500 test samples. We include Gray-Scott in the main

[1] Data source for the DiffusionPDE benchmark release: https://github.com/jhhuangchloe/DiffusionPDE. Data source of The Well: https://huggingface.co/polymathic-ai

benchmark to evaluate SBNO on a multi-channel reaction-diffusion system with pattern-forming dynamics beyond the DiffusionPDE benchmark suite.

In addition to the five main datasets, Appendix H reports smaller-scale experiments on three time-dependent PDE datasets from APEBench [13]: Korteweg-De Vries, Kuramoto-Sivashinsky, and Kolmogorov flow. These additional datasets are used only for complementary evaluation and are not included in the main benchmark tables.

Although FunDPS originally included the bounded Navier-Stokes dataset, we exclude this dataset from our experiments due to anomalies identified in the version provided by the original authors. We provide further discussion and analysis of this issue in Appendix D.3.

D.1 Time-independent Static PDEs

Following the standard dataset construction procedure used in Fourier Neural Operator (FNO) benchmarks [17], we first sample Gaussian random fields $\boldsymbol{\mu}$ on Ω and then construct the corresponding coefficient field $\mathbf{a}$. For each sample, we solve the governing equation with a second-order finite-difference scheme on the 128×128 grid. Homogeneous Dirichlet boundary conditions are enforced by applying an appropriate boundary mollifier to the solution. All three static PDEs are based on Gaussian random fields $\boldsymbol{\mu} \sim \mathcal{N}(0, (-\Delta + 9\mathbf{I})^{-2})$ defined over the spatial domain Ω.

D.1.1 Darcy Flow

Darcy flow models the steady-state flow of an incompressible viscous fluid through a porous medium. We construct a piecewise-constant permeability field $\mathbf{a}$ by thresholding a Gaussian random field: $\mathbf{a}(x) = 12$ if $\boldsymbol{\mu}(x) \geq 0$ and $\mathbf{a}(x) = 3$ otherwise. The solution $\mathbf{u}$ satisfies

$$-\nabla \cdot \big(\mathbf{a}(x)\nabla \mathbf{u}(x)\big) = f(x), \quad x \in \Omega, \tag{31}$$

with homogeneous Dirichlet boundary conditions $\mathbf{u}|_{\partial\Omega} = 0$ and $f(x) = 1$.

D.1.2 Poisson Equation

The Poisson equation describes steady-state diffusion with a source term. We set the coefficient field as $\mathbf{a} = \boldsymbol{\mu}$ and solve

$$\nabla^2 \mathbf{u}(x) = \mathbf{a}(x), \quad x \in \Omega, \tag{32}$$

under homogeneous Dirichlet boundary conditions $\mathbf{u}|_{\partial\Omega} = 0$.

D.1.3 Helmholtz Equation

The Helmholtz equation models wave propagation in heterogeneous media. We set the coefficient field as $\mathbf{a} = \boldsymbol{\mu}$ and the solution $\mathbf{u}$ satisfies

$$\nabla^2 \mathbf{u}(x) + k^2 \mathbf{u}(x) = \mathbf{a}(x), \quad x \in \Omega, \tag{33}$$

with $k = 1$ and $\mathbf{u}|_{\partial\Omega} = 0$.

D.2 Time-dependent PDEs

D.2.1 Incompressible Navier-Stokes Equation

Our time-dependent benchmarks consist of the 2D incompressible Navier-Stokes equations in an unbounded (periodic) setting on the unit torus in vorticity form. The initial vorticity field w_0 is sampled from a Gaussian random field on Ω

$$w_0 \sim \mathcal{N}\big(0,\ 7^{1.5}(-\Delta + 49\mathbf{I})^{-2.5}\big)\,. \tag{34}$$

We consider the 2D Navier-Stokes equations for a viscous incompressible fluid on the unit torus, written in vorticity form:

$$\begin{aligned} \partial_t w(x,t) + \mathbf{u}(x,t)\cdot\nabla w(x,t) &= \nu\Delta w(x,t) + q(x), && x \in \Omega,\ t \in (0,T], \\ \nabla \cdot \mathbf{u}(x,t) &= 0, && x \in \Omega,\ t \in [0,T], \\ w(x,0) &= w_0(x), && x \in \Omega, \end{aligned} \tag{35}$$

where $w = \nabla \times \mathbf{u}$ denotes the vorticity and we set the viscosity to $\nu = 10^{-3}$. The forcing term follows a fixed spatial pattern for $x = (x_1, x_2)$:

$$q(x) = \frac{1}{10}\Big(\sin(2\pi(x_1 + x_2)) + \cos(2\pi(x_1 + x_2))\Big).$$

We simulate the dynamics up to a terminal time corresponding to one second using a pseudo-spectral solver, and store the vorticity fields on a 128×128 grid.

D.2.2 Gray-Scott Equation

The Gray-Scott system describes the spatiotemporal evolution of two interacting chemical species through coupled reaction-diffusion dynamics and is widely used as a benchmark for pattern formation. Following The Well dataset [27], we consider two concentration fields $A(x,t)$ and $B(x,t)$ on a two-dimensional spatial domain, governed by

$$\frac{\partial A}{\partial t} = \delta_A \Delta A - AB^2 + f(1 - A), \qquad \frac{\partial B}{\partial t} = \delta_B \Delta B - AB^2 - (f + k)B, \tag{36}$$

where f and k denote the feed and kill rates, respectively, and δ_A, δ_B are the diffusion coefficients. In the original dataset, all simulations use fixed diffusion coefficients, the fields are defined on a uniform Cartesian grid over $[-1, 1] \times [-1, 1]$, and periodic boundary conditions are imposed. The data contain two channels corresponding to the concentrations of species A and B, and the initial conditions are generated from two families: random Fourier series and random clusters of Gaussians.

In our experiments, we use the parameter setting with $f = 0.029$ and $k = 0.057$. Each trajectory consists of 1001 time steps on a 128×128 grid, with simulation time step 1 second and stored frames separated by $\Delta t = 10$ seconds. From each trajectory, we construct one-step forecasting pairs by taking the full field at one stored time step as input and the full field at the next stored time step as output, yielding 100 input-output pairs per trajectory. Using trajectories generated from 100 initial conditions, we obtain 10,000 training samples in total. For evaluation, we apply the same construction to trajectories generated from 5 unseen initial conditions, resulting in 500 test samples.

D.2.3 Korteweg-De Vries Equation

The Korteweg–De Vries equation models nonlinear dispersive wave propagation and is a classical benchmark for studying the interplay between advection, dispersion, and dissipation in time-dependent PDEs. Following the APEBench dataset [13], we consider a scalar field $u(x,t)$ on a two-dimensional spatial domain, governed by

$$\frac{\partial u}{\partial t} = -\frac{b}{2}(\mathbf{1} \cdot \nabla)u + \xi\, \mathbf{1} \cdot \nabla^3 u - \zeta\, \mathbf{1} \cdot \nabla^4 u, \tag{37}$$

where b, ξ, and ζ control the advection, dispersive, and dissipative terms, respectively. In the original dataset, the field is defined on a uniform Cartesian grid over $[-1, 1] \times [-1, 1]$, and periodic boundary conditions are imposed. The initial condition is generated from a truncated Fourier series. In one dimension, this takes the form

$$u(t = 0, x) = o + \sum_{k=1}^{K} a_k \sin\left(k \frac{2\pi}{L} x\right) + b_k \cos\left(k \frac{2\pi}{L} x\right), \tag{38}$$

where the offset o and the Fourier coefficients are sampled uniformly, and higher-dimensional initial conditions are constructed analogously by combining modes up to cutoff K.

In our experiments, we use the parameter setting with $b = -2.0$, $\xi = -14.0$, and $\zeta = -9.0$. Each trajectory is simulated on a 160×160 grid with time step interval $\Delta t = 1$. The training set consists of 50 trajectories, each with 50 time steps, while the validation and test data are drawn from 30 trajectories, each with 100 time steps. From each trajectory, we construct one-step forecasting pairs by taking the full field at a given time step as input and the full field 10 time steps later as the target. This yields 40 input-target pairs per trajectory for training, resulting in 2,000 training samples in total. For validation and test, we construct 20 input-target pairs per trajectory from the 30 trajectories, yielding 600 samples in total, of which 100 are used for validation and 500 for testing.

D.2.4 Kuramoto-Sivashinsky Equation

The Kuramoto-Sivashinsky equation models nonlinear spatiotemporal dynamics arising from the competition between destabilizing growth, nonlinear transport, and higher-order dissipation, and is a standard benchmark for chaotic time-dependent PDEs. Following the APEBench dataset [13], we consider a scalar field $u(x,t)$ on a two-dimensional spatial domain, governed by

$$\frac{\partial u}{\partial t} = -\frac{b}{2}\|\nabla u\|^2 - \nu\nabla\cdot\nabla u - \zeta\,\mathbf{1}\cdot\nabla^4 u, \tag{39}$$

where b, ν, and ζ control the nonlinear, second-order, and fourth-order terms, respectively. In the original dataset, the field is defined on a uniform Cartesian grid over $[-1,1]\times[-1,1]$, and periodic boundary conditions are imposed. The initial condition is generated from a truncated Fourier series, the same as the Korteweg-de Vries equation.

In our experiments, we use the parameter setting with $b=-6.0$, $\nu=-1.5$, and $\zeta=-15.0$. Each trajectory is simulated on a 160×160 grid with a time step interval $\Delta t=1$. The training set consists of 50 trajectories, each with 50 time steps, while the validation and test data are drawn from 30 trajectories, each with 100 time steps. From each trajectory, we construct forecasting pairs by taking the full field at a given time step as input and the full field 10 time steps later as the target. This yields 40 input-target pairs per trajectory for training, resulting in 2,000 training samples in total. For validation and test, we construct 20 input-target pairs per trajectory from the 30 trajectories, yielding 600 samples in total, of which 100 are used for validation and 500 for testing.

D.2.5 Kolmogorov Flow

Kolmogorov flow is a standard benchmark for forced two-dimensional incompressible turbulence and is widely used to study nonlinear transport and coherent structure formation in time-dependent PDEs. Following the APEBench dataset, we consider a scalar vorticity field $u(x,t)$ on a two-dimensional spatial domain, governed by

$$\frac{\partial u}{\partial t} = -b\left(\begin{bmatrix}1\\-1\end{bmatrix}\odot\nabla(\Delta^{-1}u)\right)\cdot\nabla u + \nu\nabla\cdot\nabla u + \lambda u - k\cos\left(k\frac{2\pi}{L}y\right), \tag{40}$$

where ν denotes the viscosity, λ controls the linear term, and k determines the forcing wavenumber. In the original dataset, the field is defined on a uniform Cartesian grid over $[-1,1]\times[-1,1]$, and periodic boundary conditions are imposed. The initial condition is generated from a truncated Fourier series, as described for the Korteweg-de Vries equation above.

In our experiments, we use the parameter setting corresponding to Reynolds number 100, with $\nu=0.01$, $\lambda=-0.1$, and $k=4$. As in the Korteweg-de Vries equation, each trajectory is simulated on a 160×160 grid with time step interval $\Delta t=1$. The training set consists of 50 trajectories, each with 50 time steps, while the validation and test data are drawn from 30 trajectories, each with 100 time steps. We construct forecasting pairs in the same way as for the Korteweg-de Vries equation, by taking the full field at a given time step as input and the full field 10 time steps later as the target. This yields 40 input-target pairs per trajectory for training, resulting in 2,000 training samples in total. For validation and test, we construct 20 input-target pairs per trajectory from the 30 trajectories, yielding 600 samples in total, of which 100 are used for validation and 500 for testing.

D.3 Data Quality Issue in the Accessed Bounded Navier-Stokes Release

Upon examining the datasets provided by FunDPS, we identified data quality issues in the bounded Navier–Stokes dataset. This dataset is intended to represent 2D incompressible flow around a cylindrical obstacle in a bounded domain. However, we observed anomalies likely arising from incorrect handling of the region near the cylindrical obstacle, suggesting a flaw in the preprocessing or boundary treatment in that area. Below, we provide visualizations of the affected samples to illustrate the observed issues.

E Baseline Details

We provide additional details on the baseline methods used in Section 5. The diffusion-based baselines are DiffusionPDE [10] and FunDPS [38]. Both methods solve reconstruction problems through

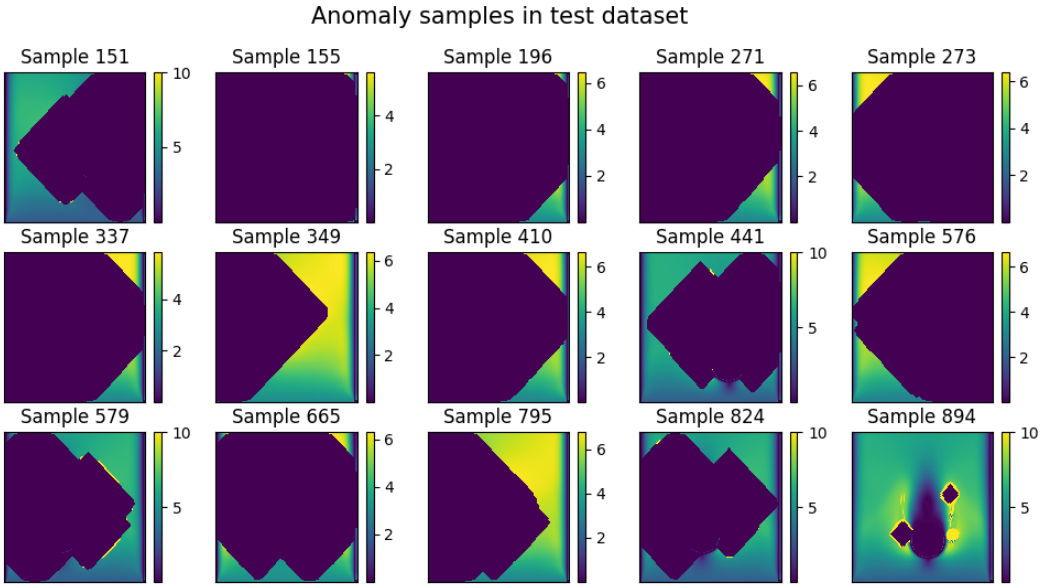


Figure 3: Visualizations of samples with data quality issue.

guided diffusion sampling, where observation consistency is imposed during inference through guidance terms rather than by concatenating an explicit binary mask channel to the model input. For FunDPS, we use its multi-resolution inference strategy in the main experiments, which uses low-resolution sampling for 80% of the inference trajectory to reduce computational cost. For completeness, we also evaluate a fixed-resolution variant of FunDPS in Appendix G.

We used the official FunDPS implementation and retrained its model directly at 128×128, following the reported architecture and training configuration for that resolution. Inference used 500 sampling steps, and only the observation and PDE guidance weights were selected using the validation split. Training was conducted on four NVIDIA B200 GPUs. Direct-resolution training should be distinguished from inference resolution: the main FunDPS results use its multi-resolution sampling strategy, whereas Appendix G.1 evaluates fixed-resolution inference.

Despite this protocol, we did not reproduce the accuracy reported in the original FunDPS paper. Our evaluation uses 5% observations, whereas its original reported setting uses 3%. Other differences include direct-128×128 training rather than the original two-stage multi-resolution training pipeline, and a different hardware and distributed-training configuration. We cannot establish which, if any, of these differences explains the discrepancy. Consequently, comparisons with FunDPS describe our reproduced configurations and should not be interpreted as establishing superiority over its originally reported performance.

For deterministic neural-operator baselines, we include FNO [17] and GNOT [8]. These two models differ in how they represent sparse observations. FNO operates on grid-structured inputs; therefore, for fair comparison with SBNO, we provide FNO with the same masked-grid representation consisting of the observed values and the corresponding binary mask. In contrast, GNOT can directly process partial observations without requiring them to be represented on the same masked grid, making it a relevant baseline for sparse-observation settings.

We also include FNOPE [23] as a probabilistic neural-operator baseline. FNOPE is particularly relevant for inverse problems because it is designed to model uncertainty in operator-learning settings. However, unlike SBNO, the neural-operator baselines considered here are not trained with a single unified objective that simultaneously supports both forward and inverse inference. Accordingly, when a baseline does not natively provide a unified forward/inverse model, we train separate models for the two tasks.

All neural-operator baselines are equation-free in the sense that they do not require access to the governing PDE or PDE residual during training or inference. The key distinction is that SBNO combines equation-free learning with a fixed empirical bridge coupling and a single trained pair

of temporal drift networks supporting both PDE tasks, whereas the deterministic and probabilistic operator baselines are trained directly for task-specific reconstruction.

F Hyperparameter Sensitivity of FunDPS in Inference

Guidance-based diffusion solvers require careful tuning of inference-time hyperparameters to achieve good reconstruction under sparse observations, in contrast to SBNO, which performs deterministic inference via the probability-flow ODE. This section reports the sensitivity trends of the guided diffusion baselines used in our experiments in Section 5.[2] To evaluate hyperparameter sensitivity, we sweep the guidance weight ζ for each baseline and measure the resulting reconstruction performance under the same evaluation protocol as in the main experiments. The quantitative comparison is presented in Table 4.

Table 4: Sensitivity of FunDPS to the inference-time guidance weight ζ under sparse observations. We report the relative ℓ_2 error for both forward and inverse tasks for five PDE benchmarks. The symbol "div" indicates that the inference procedure fails to converge.

Guidance Weight ζ	**Darcy Flow**		**Poisson**		**Helmholtz**		**Nonbounded Navier-Stokes**		**Gray-Scott**	
	Forward	Inverse	Forward	Inverse	Forward	Inverse	Forward	Inverse	Forward	Inverse
500	0.0480	0.2567	0.1649	0.4540	0.1484	0.2249	0.0624	0.1102	0.1032	0.0947
1,000	0.0374	0.1913	0.0931	0.3827	0.0725	0.2017	0.0427	0.1043	0.0920	0.0837
5,000	0.0371	0.1083	0.0775	0.2724	0.0519	0.2598	0.0358	0.2395	0.0718	0.0640
10,000	0.0449	0.0895	0.0993	0.2416	0.0499	0.4516	0.0373	0.5372	0.0662	0.0599
15,000	0.0557	0.0821	0.1324	0.2272	0.0495	0.7019	0.0426	0.4976	0.0677	0.0616
20,000	0.1254	0.0783	0.1943	0.2193	0.1201	0.9818	0.0518	0.3776	0.0695	0.0640
25,000	div	0.0763	div	0.2304	0.2594	1.2256	0.0648	0.2481	0.0712	0.0657
30,000	div	0.0751	div	0.2876	0.3199	1.3348	0.3151	0.2170	div	0.0666
40,000	div	0.0747	div	div	div	1.2531	0.5091	0.3845	div	0.1071
50,000	div	0.0742	div	div	div	1.4033	div	0.4242	div	div

Overall, both guided diffusion baselines exhibit non-negligible sensitivity to the guidance weight, and the sensitivity depends on the dataset. For sufficiently large ζ, baselines occasionally fail to converge, resulting in invalid samples or numerical divergence. We mark a run as "div" if it produces NaNs/Infs, diverges, or fails to return a valid reconstruction within the maximum inference budget. Figure 4 plots the performance as a function of the guidance weight for FunDPS. For runs that do not converge, we mark an "$\times$" at the corresponding guidance weight on the x-axis, using the same color as the associated curve.

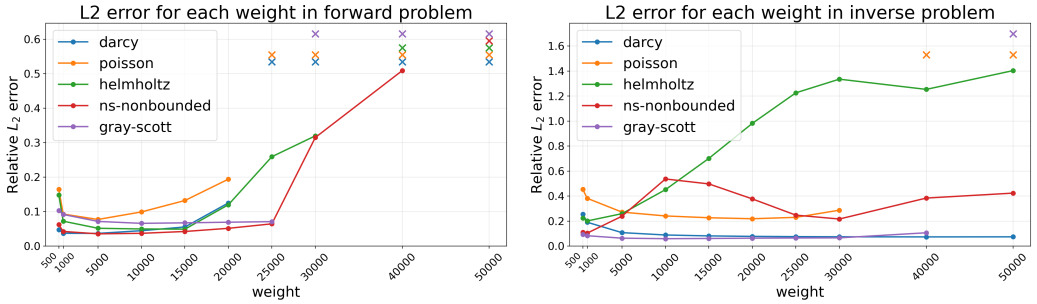


Figure 4: FunDPS sensitivity to guidance weight ζ in the forward and inverse problem. Divergent cases are marked with $\times$.

Table 4 and Figure 4 show that FunDPS is highly sensitive to the inference-time guidance weight ζ, and the degree of sensitivity varies substantially across datasets and between forward and inverse tasks. In several forward problems, performance improves as ζ increases from small values, but only up to a task-dependent range, after which the error increases again or the inference becomes

[2] We follow the same evaluation metric as Appendix C.

unstable. This trend is visible, for example, in Darcy flow, Poisson, Helmholtz, nonbounded Navier-Stokes, and Gray-Scott, where moderate guidance generally yields the best reconstruction accuracy, while overly large guidance leads to degradation or divergence. By contrast, the inverse problems often require larger guidance weights to reach their best performance, but they also tend to be more unstable, with sensitivity becoming especially pronounced in Helmholtz and nonbounded Navier-Stokes.

Another notable trend is that the optimal ζ differs significantly across problems. Forward tasks typically favor moderate guidance, whereas inverse tasks often benefit from stronger guidance but are more prone to instability. For sufficiently large ζ, inference may produce invalid samples or diverge entirely, as reflected by the "div" entries in Table 4. This reinforces the practical limitation of guidance-based diffusion solvers: their performance depends heavily on careful inference-time tuning, in contrast to SBNO, which uses a fixed deterministic inference procedure.

G Detailed Experimental Results

G.1 Fixed-resolution FunDPS

As discussed in Section 5.1, we additionally evaluate FunDPS under a fixed-resolution inference setting. We denote this variant as "FunDPS fixed" in our results. The corresponding performance is reported in Table 5, providing a direct comparison with other methods under consistent resolution settings.

Table 5: The evaluation results are reported on five benchmark datasets. Each performance metric is expressed as the "mean $\pm$ standard deviation" of the relative L_2 error across test set samples. For clarity, the best mean and best standard deviation in each column are highlighted in **bold**, while the second-best values are marked with underline.

Dataset	Problem	SBNO (Ours)	FunDPS fixed	FunDPS	DiffusionPDE	FNOPE	GNOT	FNO
Darcy flow	Forward	**0.0359 ± 0.0133**	0.0442 ± 0.0248	0.0461 ± 0.0207	0.0552 ± 0.0849	0.2219 ± 0.0893	0.0510 ± 0.0148	0.0582 ± 0.0135
	Inverse	**0.0269 ± 0.0125**	0.0742 ± 0.0348	0.0835 ± 0.0380	0.0499 ± 0.0338	0.2241 ± 0.1110	0.0779 ± 0.0367	0.0946 ± 0.0409
Poisson	Forward	**0.0424 ± 0.0219**	0.0775 ± 0.0420	0.0775 ± 0.0473	0.0986 ± 0.1696	0.7849 ± 0.4061	0.0929 ± 0.0428	0.1458 ± 0.0518
	Inverse	0.2167 ± **0.0345**	0.2193 ± 0.0609	0.3664 ± 0.1232	**0.2015** ± 0.0792	0.8423 ± 0.2244	0.3585 ± 0.0910	0.3912 ± 0.0991
Helmholtz	Forward	0.0519 ± 0.0315	**0.0495 ± 0.0266**	0.0563 ± 0.0310	0.0902 ± 0.1412	0.7388 ± 0.3684	0.0926 ± 0.0393	0.2191 ± 0.0854
	Inverse	0.2310 ± **0.0501**	0.2017 ± 0.0532	0.2550 ± 0.0740	**0.1893** ± 0.2557	0.7111 ± 0.2000	0.3728 ± 0.0865	0.4178 ± 0.0947
Nonbounded Navier-Stokes	Forward	0.0368 ± **0.0086**	**0.0358** ± 0.0109	0.0435 ± 0.0136	0.0922 ± 0.2189	1.1290 ± 0.2625	0.2360 ± 0.0476	0.2470 ± 0.0437
	Inverse	0.1097 ± **0.0167**	**0.1043** ± 0.0187	0.1241 ± 0.0234	0.1584 ± 0.2362	1.8253 ± 0.4081	0.3669 ± 0.0629	0.4106 ± 0.0557
Gray-Scott	Forward	**0.0391** ± 0.0212	0.0662 ± 0.0262	0.0846 ± 0.0222	0.0468 ± 0.0545	0.0930 ± **0.0109**	0.1025 ± 0.0084	0.0857 ± 0.0112
	Inverse	**0.0364** ± 0.0188	0.0599 ± 0.0254	0.0683 ± 0.0225	0.0589 ± 0.0622	0.1094 ± **0.0089**	0.0871 ± 0.0084	0.0724 ± 0.0102

The evaluation results show that fixed-resolution inference generally yields better performance compared to the multi-resolution setting for FunDPS, with some cases even outperforming the proposed SBNO. However, this improvement in accuracy comes at a significant computational cost: the inference time per sample increases to 4.61 seconds, which is $1.7\times$ slower than FunDPS with multi-resolution and $16.5\times$ slower than SBNO. This comparison illustrates the accuracy–cost trade-off between SBNO and the evaluated FunDPS inference configurations.

G.2 Sensitivity to Inference Seeds

To assess the variability of the reported evaluation metrics with respect to randomness in the inference procedure, we repeat SBNO inference with five different random seeds and report the resulting variation in Table 6. For each seed, we compute the relative L_2 error over the full test set and summarize the test-set error distribution by its mean and standard deviation. The table reports the average of these quantities across the five seeds together with their standard deviation across seeds.

Overall, the results show that SBNO is insensitive to the choice of inference seed. Across all datasets and both forward and inverse problems, the seed-level variation of the mean relative L_2 error is small compared with the corresponding test-set error level. These results indicate that the reported performance is not driven by a particular random seed. They also suggest that the across-instance error dispersion reported in the main results reflects variation in test-case difficulty rather than instability due to inference randomness.

Table 6: Sensitivity of SBNO to inference randomness over five different seeds. For each seed, we compute the test-set mean and standard deviation of the relative L_2 error. Each entry reports the average and standard deviation of the corresponding metric across seeds.

Problem	**Forward problem**		**Inverse problem**	
	mean	std	mean	std
Darcy Flow	0.0359 ± 0.000708	0.0133 ± 0.000627	0.0269 ± 0.000297	0.0125 ± 0.000185
Poisson	0.0424 ± 0.000595	0.0219 ± 0.000712	0.2167 ± 0.002005	0.0345 ± 0.001032
Helmholtz	0.0519 ± 0.001751	0.0315 ± 0.002282	0.2310 ± 0.003499	0.0501 ± 0.002147
Nonbounded Navier-Stokes	0.0368 ± 0.000786	0.0086 ± 0.000206	0.1097 ± 0.000403	0.0167 ± 0.000080
Gray-Scott	0.0393 ± 0.001163	0.0193 ± 0.000897	0.0382 ± 0.001187	0.0167 ± 0.000884

G.3 High-frequency Spectral Evaluation

To examine whether reconstruction accuracy is accompanied by suppression of fine-scale structure, we evaluate high-frequency spectral error on Darcy flow and Helmholtz using the same 5% observation protocol as in the main experiments. For each denormalized prediction $\hat{y}$ and ground truth y, we compute the orthonormal two-dimensional discrete Fourier transforms

$$\widehat{Y} = \mathcal{F}(\hat{y}), \qquad Y = \mathcal{F}(y).$$

Using radial discrete frequency, we define

$$\mathcal{H} = \left\{ \mathbf{k} : \|\mathbf{k}\|_2 \geq 0.5 \max_{\mathbf{j}} \|\mathbf{j}\|_2 \right\}.$$

We stabilize normalization using

$$D_{\mathrm{HF}}(y) = \max\left(\sum_{\mathbf{k}\in\mathcal{H}} |Y(\mathbf{k})|^2,\ 10^{-12} \sum_{\mathbf{k}} |Y(\mathbf{k})|^2 + 10^{-24} \right).$$

The high-frequency relative spectral error and energy ratio are

$$\mathcal{E}_{\mathrm{HF}}(\hat{y}, y) = \sqrt{\frac{\sum_{\mathbf{k}\in\mathcal{H}} |\widehat{Y}(\mathbf{k}) - Y(\mathbf{k})|^2}{D_{\mathrm{HF}}(y)}}, \quad \mathcal{R}_{\mathrm{HF}}(\hat{y}, y) = \frac{\sum_{\mathbf{k}\in\mathcal{H}} |\widehat{Y}(\mathbf{k})|^2}{D_{\mathrm{HF}}(y)}.$$

We compute both metrics per test instance and report their mean and standard deviation. All methods use the same Fourier normalization and frequency mask. Lower $\mathcal{E}_{\mathrm{HF}}$ indicates more accurate recovery of the complex Fourier coefficients. When the denominator floor is inactive, $\mathcal{R}_{\mathrm{HF}} = 1$ indicates matching high-frequency energy, while values below or above one indicate attenuation or excess energy, respectively. Since the energy ratio does not measure phase or coefficient alignment, it must be interpreted jointly with the spectral error.

Table 7: High-frequency relative spectral error under 5% observations. Values are mean $\pm$ standard deviation across test instances; lower is better.

Method	Darcy forward	Darcy inverse	Helmholtz forward	Helmholtz inverse
DiffusionPDE	2.1465 ± 2.4595	1.2648 ± 0.1078	8.9174 ± 11.6318	0.4139 ± 0.2540
FunDPS	11.1638 ± 1.7811	1.3643 ± 0.0712	4.2907 ± 1.7299	0.5651 ± 0.1460
SBNO	$\mathbf{1.6347 \pm 0.1468}$	$\mathbf{1.1543 \pm 0.0621}$	$\mathbf{1.9629 \pm 1.1171}$	$\mathbf{0.3278 \pm 0.1533}$

SBNO achieves the lowest high-frequency relative spectral error among the three evaluated methods in all four settings, reducing it relative to the next-best baseline by approximately 23.8%, 8.7%, 54.3%, and 20.8% for Darcy forward, Darcy inverse, Helmholtz forward, and Helmholtz inverse, respectively. The energy ratios do not support uniform high-frequency suppression as an explanation

Table 8: High-frequency energy ratios under 5% observations. Values are mean ± standard deviation across test instances. A ratio of one indicates matching energy when the normalization floor is inactive; the ratios should be interpreted together with spectral error.

Method	Darcy forward	Darcy inverse	Helmholtz forward	Helmholtz inverse
DiffusionPDE	10.6676 ± 25.2757	0.9310 ± 0.3236	213.7049 ± 530.5397	0.9117 ± 0.2235
FunDPS	127.4999 ± 41.0276	1.1889 ± 0.1975	22.2814 ± 22.8030	0.9577 ± 0.2244
SBNO	2.6015 ± 0.5581	0.7429 ± 0.1215	6.2776 ± 8.0286	0.9932 ± 0.5402

for these results: SBNO exhibits excess high-frequency energy in both forward tasks, although substantially less than the diffusion baselines. Some ground truths have little energy in this band, making relative energy ratios sensitive to small spurious components. Helmholtz inverse has a mean energy ratio close to one, whereas Darcy inverse exhibits attenuation. We therefore do not claim uniform spectral-energy preservation. Together, the two metrics indicate that SBNO's lower reconstruction-error dispersion cannot be attributed solely to high-frequency suppression in these evaluated settings.

G.4 Mask-channel Ablation

We assess the contribution of explicit observation-support information by independently training variants without the binary mask channel. These variants remove the mask coordinates from the lifted state and the inference ODE while retaining observed values, zero filling, randomly sampled sensor locations, the same backbone, and the 5% observation protocol. Consequently, observed zeros and zero-filled missing entries are not explicitly distinguished in the no-mask representation.

Table 9: Mask-channel ablation on all five benchmark datasets under 5% observations. Values are mean ± standard deviation of relative L_2 error across test instances. For Gray–Scott, entries average the corresponding statistics of the two reconstructed channels.

Problem	Mask-aware forward	No-mask forward	Mask-aware inverse	No-mask inverse
Darcy Flow	0.0359 ± 0.0133	0.0579 ± 0.0172	0.0269 ± 0.0125	0.0297 ± 0.0138
Poisson	0.0424 ± 0.0219	0.3339 ± 0.1829	0.2167 ± 0.0345	0.3950 ± 0.1254
Helmholtz	0.0519 ± 0.0315	0.2411 ± 0.1428	0.2310 ± 0.0501	0.3334 ± 0.1149
Nonbounded Navier-Stokes	0.0368 ± 0.0086	0.1471 ± 0.0238	0.1097 ± 0.0167	0.2267 ± 0.0243
Gray-Scott	0.0393 ± 0.0193	0.0448 ± 0.0248	0.0382 ± 0.0167	0.0385 ± 0.0210

The mask-aware model has lower mean reconstruction error in all ten settings. On Darcy flow, removing the mask increases mean error by 61.3% for forward reconstruction and 10.4% for inverse reconstruction. Larger differences are observed on Poisson, Helmholtz, and nonbounded Navier–Stokes, whereas the difference for Gray–Scott inverse is small. These results support the utility of explicitly representing observation support, beyond supplying zero-filled values alone. The mask distinguishes genuine observed zeros from missing entries, but does not eliminate the non-uniqueness of reconstruction from partial observations.

G.5 PDE-task Coupling Ablation

We distinguish the direction of the PDE reconstruction task from the temporal direction of the bridge dynamics. The forward-task coupling $\Pi^F_{0,T}$ is induced by $(\mathcal{E}_s(a), \mathcal{E}_1(u))$, whereas the inverse-task coupling $\Pi^I_{0,T}$ is induced by $(\mathcal{E}_1(a), \mathcal{E}_s(u))$. We train Darcy variants using only $\Pi^F_{0,T}$, only $\Pi^I_{0,T}$, or their equal mixture. The dataset, backbone, 5% observation setting, and evaluation protocol are otherwise unchanged. Both temporal drift networks are retained in every variant because the probability-flow ODE requires both drifts. Thus, task-specific training excludes one PDE-task coupling rather than one temporal network.

Table 10: Mask-channel ablation on all five benchmark datasets under 5% observations. Values are mean ± standard deviation of relative L_2 error across test instances. For Gray–Scott, entries average the corresponding statistics of the two reconstructed channels.

Training coupling	Forward error	Inverse error
Unified mixture	0.0359 ± 0.0133	0.0269 ± 0.0125
Forward-task coupling only	$\mathbf{0.0315 \pm 0.0073}$	0.5063 ± 0.0525
Inverse-task coupling only	0.5420 ± 0.0269	$\mathbf{0.0150 \pm 0.0072}$

Task-specific training reduces mean error on the retained task by approximately 12.3% for forward reconstruction and 44.2% for inverse reconstruction relative to unified training. However, error on the excluded task increases substantially: forward-task-only training yields an inverse error of 0.5063, while inverse-task-only training yields a forward error of 0.5420. These results reveal a specialization–unification trade-off. In this Darcy experiment, adding the opposite PDE-task coupling does not improve the retained task's point-reconstruction accuracy. The benefit of unified training is that one checkpoint supports both PDE tasks, whereas specialization improves the selected task at the expense of the unsupported capability.

G.6 Forward-drift-only SDE Evaluation

The two temporal drift networks have separate parameters, losses, optimizers, and exponential moving averages. Neither network uses the other network's outputs or parameters during optimization. Therefore, the forward drift extracted from a unified checkpoint has the same training objective as a forward drift trained alone on the same coupling mixture, although separately executed training runs need not yield identical parameters.

To evaluate the forward drift without the backward drift, we use forward SDE inference, since our probability-flow ODE requires both networks. We compare the forward drift trained on the unified mixture with that trained only on the forward-task coupling. Each test case uses 60 stochastic paths, with identical observation masks and Brownian seeds across the two variants. We evaluate the sample-mean reconstruction and report the mean and standard deviation of its relative L_2 error across test instances.

Table 11: Mask-channel ablation on all five benchmark datasets under 5% observations. Values are mean ± standard deviation of relative L_2 error across test instances. For Gray–Scott, entries average the corresponding statistics of the two reconstructed channels.

Forward-drift training coupling	Inference	Forward error
Unified mixture	Forward-drift-only SDE	0.0263 ± 0.0064
Forward-task coupling only	Forward-drift-only SDE	$\mathbf{0.0174 \pm 0.0056}$

Specializing the forward drift to the forward-task coupling reduces the mean error by approximately 34%. Consistent with the ODE coupling ablation, this matched SDE comparison does not show an accuracy benefit from adding the opposite-task coupling on Darcy forward reconstruction. Unified training instead provides both PDE capabilities when the two learned drifts are used together. Because this table evaluates a 60-path sample mean, its error values should not be interpreted as a comparison of individual SDE samples with deterministic ODE predictions.

H Additional Experiments on Diverse Datasets

In this section, we report additional experimental results on three time-dependent datasets that were not included in the main text: Korteweg–De Vries, Kuramoto–Sivashinsky, and Kolmogorov flow. Details of these datasets are provided in Appendix D. Unless otherwise stated, all experimental settings, training protocols, and evaluation procedures are the same as those used in Section 5.

We separate these experiments from the main text because their scale is substantially smaller than that of the primary benchmarks in Section 5. In particular, these datasets contain fewer trajectories and fewer total training pairs, making them less suitable as the main basis for large-scale quantitative comparison. Nevertheless, they provide a useful complementary testbed, since they exhibit different forms of nonlinear and turbulent dynamics from those considered in the main experiments. In this sense, they offer additional evidence on whether SBNO can remain effective beyond the specific PDE families emphasized in Section 5.

Table 12: Evaluation results on three additional APEBench datasets. Each performance metric is reported as the "mean $\pm$ standard deviation" of the relative L_2 error across test samples. For clarity, the best mean and best standard deviation in each row are highlighted in **bold**, while the second-best values are marked with underline.

Dataset	Problem	SBNO (Ours)	FunDPS	DiffusionPDE	FNOPE	GNOT	FNO
Korteweg	Forward	0.2858 $\pm$ 0.0650	1.5024 $\pm$ 0.2236	**0.2177** $\pm$ 0.2017	0.8498 $\pm$ 0.0446	1.1628 $\pm$ 0.0692	0.5356 $\pm$ **0.0348**
–De Vries	Inverse	0.2866 $\pm$ 0.1005	1.5401 $\pm$ 0.2491	**0.1765** $\pm$ 0.0791	0.9399 $\pm$ 0.0571	1.1798 $\pm$ 0.0739	0.7483 $\pm$ **0.0202**
Kuramoto–	Forward	**0.3236** $\pm$ **0.0295**	1.2990 $\pm$ 0.0754	0.4517 $\pm$ 0.0992	1.0500 $\pm$ 0.0711	1.0707 $\pm$ 0.0579	0.6808 $\pm$ 0.0359
Sivashinsky	Inverse	**0.3674** $\pm$ **0.0283**	1.2962 $\pm$ 0.0733	0.3973 $\pm$ 0.0584	1.0451 $\pm$ 0.0716	1.0660 $\pm$ 0.0568	0.6727 $\pm$ 0.0389
Kolmogorov	Forward	**0.5322** $\pm$ 0.0978	1.2445 $\pm$ **0.0560**	0.5551 $\pm$ 0.1133	0.6180 $\pm$ 0.0882	0.9275 $\pm$ 0.0860	0.6211 $\pm$ 0.0764
Flow	Inverse	**0.4849** $\pm$ 0.0810	1.2329 $\pm$ 0.0525	0.5739 $\pm$ 0.2634	0.5687 $\pm$ 0.0632	0.9071 $\pm$ 0.0814	0.6039 $\pm$ **0.0466**

Table 12 summarizes the reconstruction results under the same partial-observation setting as in the main paper. Overall, SBNO remains competitive across all three datasets and achieves the best mean error on four out of the six forward/inverse tasks, namely both Kuramoto-Sivashinsky tasks and both Kolmogorov flow tasks. On Korteweg-de Vries, SBNO achieves the second-best mean error in both directions, while DiffusionPDE performs best. This trend is consistent with our observations in some of the more ill-posed settings in the main text: PDE-guided diffusion baselines can benefit from additional physics-based constraints at inference time, whereas SBNO is intentionally equation-free. Even in these cases, however, SBNO remains substantially more accurate than the other deterministic and probabilistic operator baselines considered here.

Another notable trend is that SBNO often combines competitive mean accuracy with relatively low dispersion, although the smallest standard deviation is not always achieved by SBNO. In several cases, FNO attains a smaller error standard deviation despite having a clearly worse mean error, indicating more concentrated but systematically less accurate predictions. By contrast, the diffusion baselines sometimes exhibit strong average performance but also larger variability across test samples. Taken together, these results suggest that SBNO continues to provide a favorable trade-off between reconstruction accuracy and test-instance error dispersion even on smaller-scale datasets with distinct dynamical characteristics.

I Uncertainty Quantification with Stochastic Sampling

Although the main paper focuses on deterministic inference via the probability-flow ODE for efficiency and stable deterministic point reconstruction, the learned fixed-coupling bridge dynamics in SBNO can also be used in a stochastic sampling mode for uncertainty quantification. To examine this capability more broadly, we additionally perform stochastic-sampling experiments for all datasets and for both forward and inverse problems under the same partial-observation setting as in the main experiments. Specifically, for each test case, we draw 60 stochastic samples from the learned bridge dynamics and report their empirical predictive statistics in Table 13, without assuming exact recovery of the target conditional law.

We organize the reported quantities into four groups. First, under "sample-mean reconstruction", we evaluate the accuracy of the sample mean prediction using the relative L_2 error mean and standard deviation across test samples, together with the mean absolute L_1 error. These metrics indicate whether stochastic sampling preserves accurate reconstruction when the posterior is summarized by its mean. Second, under "uncertainty magnitude", we report the global mean sample standard deviation and variance, which quantify the overall scale of predictive uncertainty across the spatial domain and test set.

Third, under "calibration and sharpness", we report empirical coverage and interval width at the 50%, 80%, and 90% empirical predictive intervals. The empirical coverage measures how often the ground-truth field values fall inside the corresponding posterior interval, while the interval width

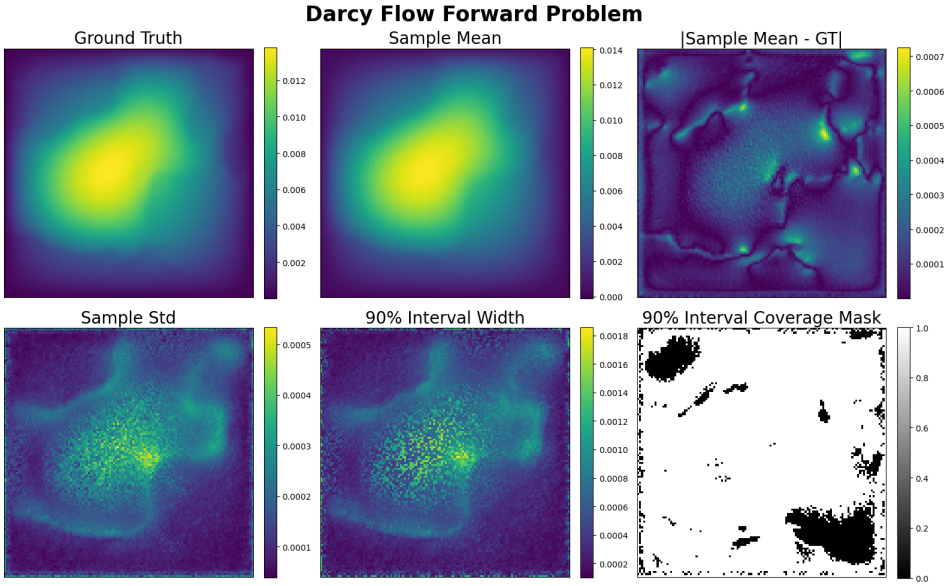
Table 13: Uncertainty-quantification results of SBNO using stochastic posterior sampling. For each dataset and for both forward and inverse problems, we draw 60 stochastic bridge samples per test case and report metrics for sample-mean reconstruction, uncertainty magnitude, calibration and sharpness, and the relation between predictive error and estimated uncertainty.

Category	Metric	**Darcy Flow**		**Poisson**		**Helmholtz**		**Nonbounded Navier-Stokes**		**Gray-Scott**	
		Forward	Inverse	Forward	Inverse	Forward	Inverse	Forward	Inverse	Forward	Inverse
Sample mean reconstruction	Rel L_2 error mean	0.0263	0.1296	0.0519	0.1271	0.0539	0.1362	0.0717	0.1342	0.0254	0.0272
	Rel L_2 error std	0.0064	0.0334	0.0210	0.0361	0.0297	0.0362	0.0072	0.0182	0.0119	0.0162
	Abs L_1 error mean	0.0001	0.2217	0.0001	0.0257	0.0002	0.0260	0.0142	0.0267	0.0089	0.0037
Uncertainty magnitude	Global mean std	0.0002	0.2016	0.0001	0.0144	0.0001	0.0131	0.0128	0.0172	0.0116	0.0060
	Global mean var	0.0000	0.6258	0.0000	0.0002	0.0000	0.0002	0.0002	0.0003	0.0002	0.0001
Calibration and sharpness	50% Coverage	0.2826	0.9858	0.2601	0.2170	0.2438	0.2025	0.1895	0.1602	0.3946	0.5085
	80% Coverage	0.5425	0.9903	0.4949	0.3923	0.4632	0.3712	0.4078	0.3150	0.6617	0.8169
	90% Coverage	0.6987	0.9923	0.6241	0.4806	0.5843	0.4574	0.6256	0.4679	0.7697	0.9136
	50% Width	0.0001	0.1524	0.0001	0.0181	0.0001	0.0167	0.0100	0.0141	0.0117	0.0062
	80% Width	0.0003	0.2985	0.0002	0.0350	0.0002	0.0321	0.0212	0.0292	0.0246	0.0128
	90% Width	0.0005	0.4012	0.0004	0.0451	0.0003	0.0413	0.0362	0.0491	0.0336	0.0174
Error-uncertainty relation	Corr(error, std)	0.2362	0.6181	0.4778	0.2103	0.0908	0.2315	0.0362	0.0814	0.4720	0.5787

reflects the sharpness of the uncertainty estimate. Taken together, these metrics characterize whether the empirical predictive distribution is both informative and appropriately calibrated: wider intervals may improve coverage but reduce sharpness, whereas overly narrow intervals may indicate overconfident uncertainty estimates.

Finally, under "error–uncertainty relation", we report the correlation between the absolute error of the sample mean and the sample standard deviation. This correlation measures the spatial association between reconstruction error and sample variability; its strength varies substantially across datasets and tasks.

Overall, these metrics provide a complementary view to the deterministic results in the main paper. Deterministic probability-flow ODE inference remains preferable when a fast and stable deterministic point reconstruction is desired, whereas stochastic bridge sampling enables empirical predictive-uncertainty estimates that can be evaluated in terms of reconstruction quality, uncertainty magnitude, calibration, and error awareness. For illustration, Figure 5 visualizes a representative example of the empirical predictive distribution, including the sample mean, sample standard deviation, interval width, and empirical coverage mask.



In figure 6, we compare ODE reconstruction with an individual SDE sample on representative Darcy-flow and nonbounded Navier–Stokes forward cases. In these examples, ODE inference re-

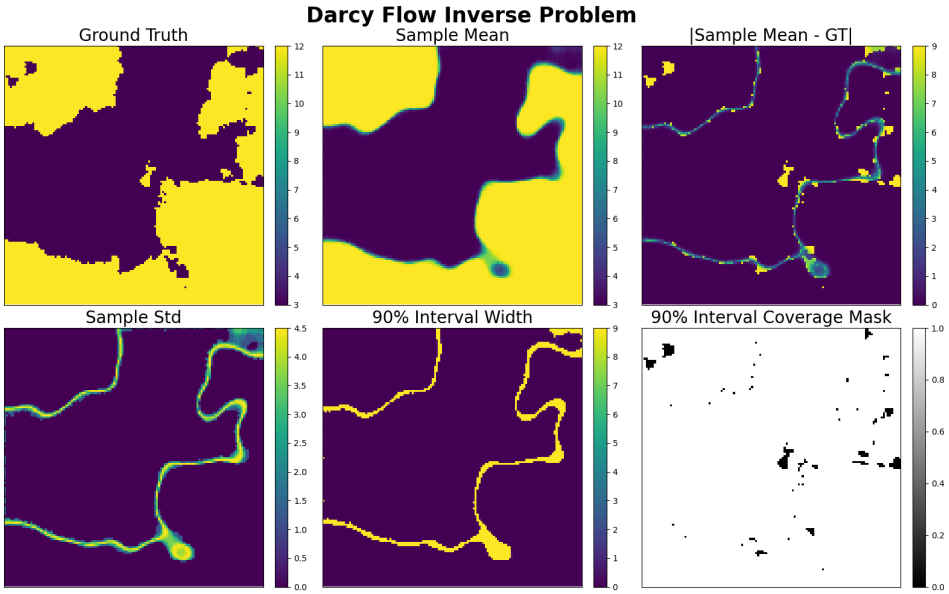


Figure 5: Representative uncertainty-quantification example of SBNO under partial observations using stochastic posterior sampling. We draw 60 stochastic bridge samples and visualize the ground truth, sample mean, absolute error of the sample mean, sample standard deviation, 90% posterior interval width, and the corresponding 90% empirical coverage mask. The sample mean summarizes the reconstruction, while the posterior spread highlights spatial regions associated with larger predictive uncertainty.

produces the dominant spatial structures, while individual SDE predictions retain the overall pattern but exhibit more diffuse features and reduced sharpness. The individual sample should be distinguished from the 60-sample mean evaluated in Table 13. These visualizations illustrate the behavior of the two inference modes on selected cases; they do not establish calibrated uncertainty or exact recovery of the target conditional distribution.

J Robustness to Noisy Partial Observations

We additionally evaluate the robustness of SBNO under noisy partial observations, which are common in practical measurement settings. To this end, we conduct additional experiments on Darcy flow by injecting Gaussian noise into the observed partial inputs at multiple noise levels. We compare SBNO against the diffusion-based baselines under the same protocol, and the corresponding results are summarized in Table 14.

Table 14: Evaluation results with Gaussian noise added to partial observations. Noise std denotes the standard deviation of the Gaussian noise. For clarity, the best mean in each column is highlighted in **bold**, while the second-best values are marked with underline.

Noise std	Problem	SBNO (Ours)	FunDPS	DiffusionPDE	GNOT
0.1	Forward	**0.0359**	0.0461	0.0371	0.0510
	Inverse	**0.0269**	0.0834	0.0743	0.1031
0.5	Forward	**0.0359**	0.0461	0.0372	0.1709
	Inverse	**0.0274**	0.0837	0.0749	0.4092
1.0	Forward	**0.0358**	0.0461	0.0373	0.2657
	Inverse	**0.0291**	0.0838	0.0766	0.4764
3.0	Forward	**0.0363**	0.0461	0.0382	0.3389
	Inverse	**0.0448**	0.0861	0.0851	0.5011

As shown in Table 14, SBNO remains competitive across all tested noise levels in both the forward and inverse problems. Under low to moderate noise, the reconstruction quality changes only

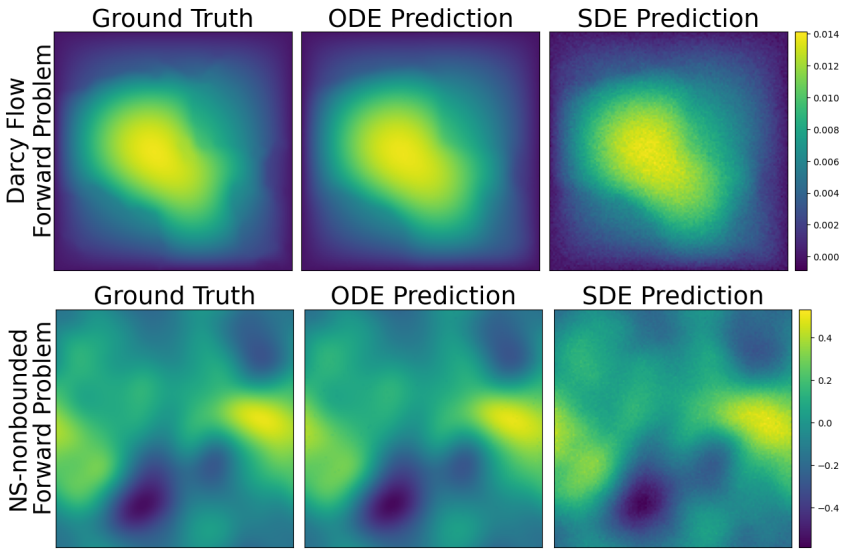


Figure 6: Comparison of ODE and SDE inference on representative Darcy-flow and nonbounded Navier–Stokes forward cases under 5% observations. For each case, we show the ground truth, deterministic ODE reconstruction, one stochastic SDE sample, and the mean of 60 SDE samples. All predictions use the same observation mask, and panels within each case share a color scale. The individual SDE sample is shown separately from the sample-mean reconstruction.

slightly, indicating that the model is reasonably stable to imperfect observations in this regime. As the noise level increases further, performance degrades as expected, with the effect being more noticeable in the inverse problem than in the forward problem. This trend is consistent with the greater ill-posedness of inverse inference under sparse and corrupted observations. Nevertheless, SBNO continues to compare favorably with the considered baselines throughout the evaluated range.

These results strengthen the empirical evaluation by showing that SBNO is not limited to the clean-observation setting. At the same time, they also indicate that robustness under noisy measurements remains an important dimension of the problem, especially in more challenging inverse settings. Other forms of measurement corruption and non-Gaussian noise remain important directions for future work.

K Sensitivity to the ODE Solver and Number of Inference Steps

We additionally investigate whether the performance gap on the more challenging inverse tasks can be reduced by using a more accurate ODE solver configuration at inference time. To this end, we evaluate SBNO with increased numbers of RK4 steps and also consider an adaptive DOPRI solver. We focus on the two inverse settings where SBNO did not outperform DiffusionPDE in the main experiments. The corresponding reconstruction performance and per-sample inference time are summarized in Table 15.

Table 15: Reconstruction performance under different ODE solver configurations on the challenging inverse tasks where SBNO did not outperform DiffusionPDE in the main text, together with the corresponding per-sample inference time. We increase the number of RK4 integration steps and additionally evaluate an adaptive DOPRI solver to assess whether improved numerical integration can close the remaining performance gap.

Step number	32	64	96	128	dopri
Poisson Inverse	0.2167	0.2074	0.2072	0.2073	0.2166
Helmholtz Inverse	0.2310	0.2220	0.2243	0.2249	0.2310
Time per Sample(sec)	0.28	0.56	0.83	1.10	0.57

As shown in Table 15, increasing the number of ODE solver steps leads to a modest improvement over the default inference configuration, indicating that part of the error can be reduced through finer numerical integration. However, the gains quickly saturate as the step count increases, and the overall improvement remains limited relative to the additional inference cost. A similar trend is observed with the adaptive solver, which does not provide a meaningful advantage over the fixed-step configurations. Overall, these results suggest that the remaining performance gap on these tasks is not primarily caused by insufficient numerical accuracy of the probability-flow ODE solver.

This observation is consistent with our explanation for the challenging inverse settings. In these cases, the dominant difficulty appears to come from the ambiguity of the inverse mapping under sparse observations rather than from discretization error along the deterministic flow trajectory. As a result, simply refining the deterministic solver is not sufficient to close the gap. This analysis clarifies that the default inference setting provides a practical trade-off between reconstruction quality and efficiency, while more aggressive solver settings yield only limited additional benefit.

L Sensitivity to the Observation Ratio

We additionally evaluate how SBNO behaves when the amount of available partial observation at inference time differs from the 5% regime used in the main experiments. To this end, we vary the observation ratio at test time while keeping the training setup unchanged, and report the corresponding reconstruction performance in Table 17.

Table 16: Evaluation results with varying ratios of partial observations. The observation ratio denotes the proportion of points in the full grid that are provided as partial observations for evaluation.

Observation ratio	**5%**	**4%**	**3%**	**2%**	**1%**	**0.5%**
Forward	0.0359	0.0402	0.0749	0.1226	0.2212	0.2727
Inverse	0.0269	0.0498	0.0865	0.2107	0.2924	0.3362

As shown in Table 17, SBNO performs best at the 5% observation ratio used during training, and the reconstruction error increases as the number of observed points decreases. This trend is visible in both the forward and inverse problems, with the degradation becoming more pronounced in the lower-observation regime. In particular, once the observation ratio falls substantially below the training condition, performance drops more sharply, especially for the inverse problem.

This behavior is consistent with a shift in the conditioning distribution induced by changing the observation ratio. In the current setup, SBNO is optimized using masked observations sampled to match the target sparse setting in the main experiments, namely 5% observations. When inference is instead performed with substantially fewer observations, the model must extrapolate to a much weaker conditioning signal than it was exposed to during training. As a result, the learned deterministic reconstruction becomes less reliable as the available information decreases.

These results clarify an important limitation of the current formulation. While SBNO is effective in the sparse-observation regime it is trained for, its performance is not yet robust across substantially different observation ratios without retraining or additional adaptation.

M Dynamic-ratio Training

Dynamic-ratio training. To examine whether exposure to varying conditioning densities improves robustness, we train a separate Darcy model with $r \sim \mathrm{Uniform}(0.5\%, 10\%)$ at each training iteration, while also resampling sensor locations. We then evaluate the same checkpoint across the trained ratio range without retraining or adaptation.

Table 17: Evaluation results with varying ratios of partial observations. The observation ratio denotes the proportion of points in the full grid that are provided as partial observations for evaluation.

Observation ratio	**10%**	**8%**	**5%**	**4%**	**3%**	**2%**	**1%**	**0.5%**
Forward	0.0664	0.0703	0.0836	0.0880	0.0954	0.1112	0.1435	0.1790
Inverse	0.0442	0.0437	0.0474	0.0511	0.0595	0.0773	0.1240	0.1710

Forward error decreases throughout the evaluated range, while inverse error decreases and approximately plateaus at 8%–10%. Compared with the fixed-density model, dynamic training reduces forward/inverse errors from 0.2727/0.3362 to 0.1790/0.1710 at 0.5%, and from 0.1266/0.0810 to 0.0664/0.0442 at 10%. However, at the specialized 5% condition, the fixed-density model remains more accurate, with errors of 0.0359/0.0269 compared with 0.0836/0.0474. These results demonstrate a specialization–robustness trade-off: fixed-density training favors a known observation regime, whereas dynamic-ratio training accommodates varying sensor availability within the evaluated range. We have not evaluated the dynamic model at 15% or 20%, so these results do not establish robustness beyond its training range.

N Closed-loop Forward/Inverse Evaluation

To examine recursive use, we perform an evaluation-only closed-loop experiment on held-out pairs with independently sampled observation masks, without closed-loop training. Specifically, after obtaining a prediction in one direction, we reuse that prediction as the partial observation for inference in the opposite direction. Concretely, for forward inference, we use the prediction from backward inference as the partial input observation, and for backward inference, we use the prediction from forward inference as the partial solution observation. Training uses clean partial fields rather than model-generated feedback, and includes neither closed-loop supervision nor a cycle-consistency objective. We did not evaluate training-set closed-loop performance, so the reported results do not establish greater consistency on training pairs. We evaluate this closed-loop setting on Darcy flow and Helmholtz. We choose Darcy flow because SBNO performs strongly in both directions on this dataset and Helmholtz because it provides a more challenging case, particularly for the inverse problem. The evaluation results are presented in Table 18. We denote the resulting reconstructions as SBNO (loop).

Table 18: Evaluation results for the closed-loop experiment, where inference is run again in the opposite direction using the model prediction from the first pass as the new partial observation. For forward inference, the prediction obtained from backward inference is used as the partial input observation. Conversely, for backward inference, the prediction obtained from forward inference is used as the partial solution observation. The resulting performance is reported as "SBNO (loop)."

Dataset	**Problem**	**SBNO**	**SBNO (loop)**
Darcy flow	Forward	0.0359 ± 0.0133	0.0369 ± 0.0132
	Inverse	0.0269 ± 0.0128	0.0531 ± 0.0216
Helmholtz	Forward	0.0519 ± 0.0315	0.1886 ± 0.1069
	Inverse	0.2310 ± 0.0501	0.2506 ± 0.0722

On Darcy flow, the forward problem remains largely stable under looping, with the relative L_2 error changing only slightly. In contrast, the inverse problem shows a more noticeable degradation. On Helmholtz, the degradation is more pronounced in the forward direction, while the inverse direction changes more moderately.

Overall, the closed-loop experiment indicates that SBNO preserves a meaningful degree of directional compatibility, but does not fully eliminate recursive error accumulation. In particular, once model predictions are reused as observations, they no longer behave like clean partial measurements and instead resemble noisy or smoothed surrogates of the true field, already containing approximation error and attenuated high-frequency structure. As a result, repeated forward/inverse reuse tends to produce smoother reconstructions with reduced fine details, together with increased reconstruction error in the harder settings. This can be seen in the Figure 7. These observations indicate limited directional compatibility in the unified model, with recursive reconstruction remaining sensitive to errors in the intermediate prediction. An extension could incorporate model-generated feedback or cycle-consistency training while retaining randomly sampled observation masks, although its effect remains to be evaluated.

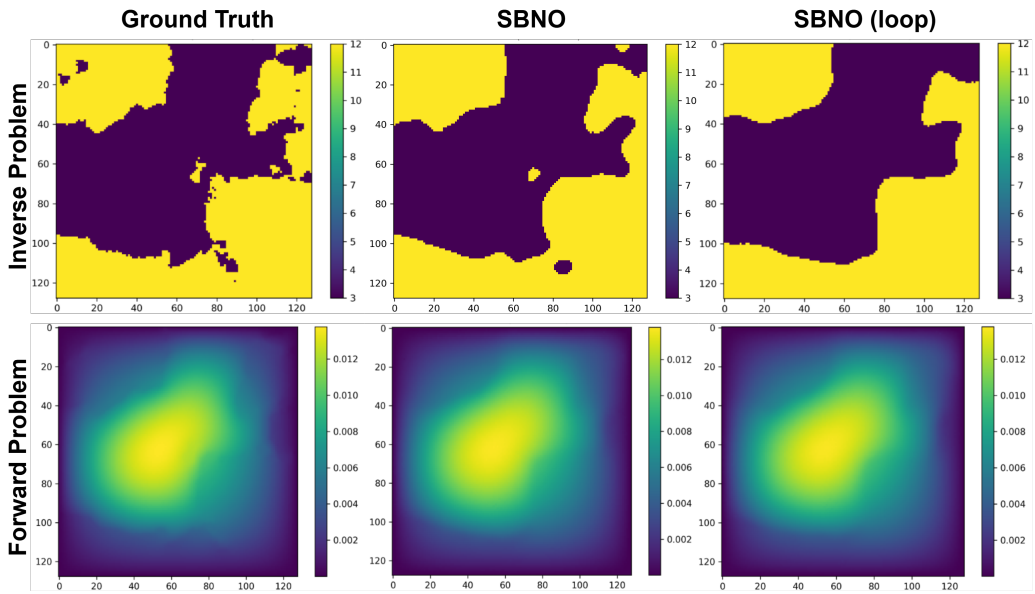


Figure 7: **Representative closed-loop visualization**. We show the original SBNO predictions and the reconstructions obtained after reusing those predictions as partial observations for inference in the opposite direction.

O Experimental Details

O.1 Reproducibility

We are committed to ensuring the reproducibility of our research. All experimental procedures, data sources, and algorithms used in this study are clearly documented in the paper. The code and datasets will be provided as the supplementary material and be made publicly available upon publication, allowing others to validate our findings and build upon our work.

O.2 Environments

All models in the experiments of Section 5 are trained for 200 epochs on four NVIDIA B200 GPUs. All inference results for these experiments are obtained using a single NVIDIA B200 GPU. For the additional experiments in Appendix H, all models are trained for 100 epochs on four NVIDIA A6000 GPUs. Details of the Python version and the specific packages used in the experiments are provided in the environment file included with the accompanying code.

O.3 Hyperparameters

In this section, we describe the hyperparameter settings used for each model. The best hyperparameters of all models were selected based on the validation split, by choosing the configuration that achieves the lowest relative L_2 error. For DiffusionPDE and FunDPS, we follow the hyperparameter choices reported in the original papers for model training and architecture and only search the inference-time guidance weight ζ. The candidate guidance weights are those reported in Appendix F. For SBNO, we use 32 RK4 steps by default, 128 ODE right-hand-side evaluations, corresponding to 256 individual drift-network evaluations. Table 19 summarizes the hyperparameters used in all experiments, including the selected guidance weight for each dataset and problem. Unless stated otherwise, all other hyperparameters are fixed across datasets, and the same validation protocol is applied to ensure a fair and reproducible comparison.

Table 19: Best hyperparameters for each model used in the main experiments.

Models	Hyperparameter	**Darcy Flow**		**Poisson**		**Helmholtz**		**Nonbounded Navier-Stokes**		**Gray-Scott**	
		Forward	Inverse	Forward	Inverse	Forward	Inverse	Forward	Inverse	Forward	Inverse
Ours	learning rate	0.0001									
	ODE solver step	32									
	Sigma min	0.2		0.001		0.001		0.001		0.001	
	Sigma max	0.2		0.3		0.3		0.45		0.3	
FunDPS multi-res	Observation guidance	1,000	25,000	5,000	10,000	10,000	5,000	5,000	1,000	5,000	10,000
	PDE guidance	0	0	0	0	1	1	1	10	0	0
FunDPS	Observation guidance	5,000	50,000	5,000	20,000	15,000	1,000	5,000	1,000	10,000	10,000
	PDE guidance	0	0	0	0	1	1	1	10	0	0
DiffusionPDE	Observation guidance	500	1500	200	300	300	200	500	500	100	100
	PDE guidance	1000	100	100	1	100	1	100	100	100	100
FNOPE	Layers number	6									
	Hidden dimension	128									
	Head number	4									
GNOT	Layers number	4									
	Hidden dimension	128									
FNO	Layers number	6									
	Hidden dimension	128									

P Impact Statement

This work follows standard ethical guidelines for scientific research. SBNO is designed for PDE operator learning under sparse or missing observations and does not involve human-subject data or personal information. We do not see a clear direct path from deploying our method to harmful uses such as surveillance, disinformation, or privacy attacks. We expect that improved forward/inverse PDE modeling can benefit applications like climate and geophysical simulation, biomedical modeling, and engineering analysis where observations are limited. As with most advances in simulation and inverse modeling, there is a general dual-use possibility, so high-stakes deployments should include appropriate validation and governance.

All experiments were conducted in compliance with relevant laws and ethical guidelines, ensuring the integrity of our findings. We are committed to transparency and reproducibility in our research processes. The code will be provided as the supplementary material and be made publicly available upon publication, allowing others to validate our findings and build upon our work.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The findings in Section 5 and Appendix support and are consistent with the main claims articulated in the abstract and introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We explicitly address the limitations of our work in Section 6, where we discuss key assumptions and potential weaknesses of our approach.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: This paper does not include any theorems, and the theoretical foundations of the model are discussed in Section 3 and Appendix A.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We provide detailed information of hyperparameters for each model in Appendix O.3 and outline the full training procedure in the algorithm presented in Appendix B. We also discuss reproducibility in Appendix O.1

Guidelines:

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

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

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

   Answer: [Yes]

   Justification: To ensure reproducibility, we include the full experimental code in the supplementary material, accompanied by a README file detailing the data generation process and step-by-step instructions for reproducing all main results.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Comprehensive information about the datasets, including their sizes and the hyperparameters used in each experiment, is available in Appendix D and O.3.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Table 2 reports results for the experiment in Section 5, including standard deviations of each metric to capture variability across samples. Additionally, Table 6 includes standard deviations of results for five seeds.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix O.2 includes details on hardware used. Additionally, we provide a comparison of computational costs in Section 5 including runtime, and memory requirements for each experimental run.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We confirm that our work adheres to the NeurIPS Code of Ethics in all respects, including the preservation of anonymity and ethical considerations regarding data usage and experimental design.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

Justification: In Section P, we address both potential positive and negative societal impacts of our work. We commit to ongoing monitoring of broader impacts and support reproducibility through open code and evaluation protocols.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper poses no such risks.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: Among the datasets used in our experiments, four datasets were previously utilized by Huang et al. [10], Yao et al. [38] and Gray-scott dataset was introduced by The Well [27]. In Appendix D, we provide the source from which we obtained this dataset, along with the original paper and a corresponding URL.

Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: In the supplementary material, we provide the code used for the experiments described in Section 5, along with documentation that guides readers through the experimental process. All code has been properly anonymized for submission.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve research with human subjects.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

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