Posterior-First Neural PDE Simulation: Inferring What Downstream Tasks Need from a Single Field

Wenshuo Wang[1], **Fan Zhang**[2,3,4*]
[1] School of Future Technology, South China University of Technology, China
[2] State Key Laboratory of Ocean Sensing,
ZJU-Hangzhou Global Scientific and Technological Innovation Center,
Zhejiang University, Hangzhou, 311215, China
[3] Ocean College, Zhejiang University, Zhoushan, Zhejiang 316000, China
[4] Kavli Institute for Astrophysics and Space Research,
Massachusetts Institute of Technology, Cambridge, MA, 02139, USA
202364870251@mail.scut.edu.cn, f.zhang@zju.edu.cn

Abstract

Neural PDE simulators often receive only a single observed field at deployment. In this setting, a field-to-future predictor can collapse distinct latent problem states into the same deterministic interface, losing the ambiguity needed for reliable rollout and downstream decisions. We propose **posterior-first neural PDE simulation**, a task-to-target-to-interface paradigm: the declared downstream tasks determine the minimal task-sufficient problem state to infer, and the posterior over that state conditions the rollout and downstream queries. This posterior can be supplied by amortized posterior estimators, including simulation-based inference with learned summaries; our contribution is the target, the interface, and their theory. We show that Bayes downstream values factor through this posterior, proper scoring rules on refinement labels identify it as their population-risk minimizer, and deterministic collapse incurs an ambiguity barrier whenever the true posterior is non-Dirac. Synthetic exact-ambiguity experiments show that point-versus-posterior gaps track the predicted barrier. On metadata-hidden PDEBench tasks, posterior recovery reduces pooled rollout nRMSE from 0.175 to 0.132, closing 59.4% of the direct-to-oracle gap. Matched controls attribute the gains to task-aligned targeting and posterior deployment, and the gains persist with a latent-diffusion backbone. These results suggest that single-observation neural PDE simulation should be posterior-first rather than monolithic field-to-future prediction.

1 Introduction

Across science and engineering, many central problems in forecasting, design, and control require fast simulation of systems governed by partial differential equations. Neural PDE simulators have therefore become an important surrogate modeling paradigm, because they can amortize the cost of repeated numerical solves [1–3]. We focus on the single-observation regime, where the simulator receives one field and must predict what happens next. This regime matters because a single observed field is often the practical input available at deployment, while the simulator is still expected to support long-horizon prediction and other downstream scientific use. But this regime is fundamentally difficult. One field need not uniquely determine the future, because hidden state, coefficients, operators, boundary conditions, or memory may remain unresolved [4–7].

*Corresponding author.
Our code is available at https://github.com/VinceWangVW/Posterior-First.

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

Existing neural PDE simulators differ in what they receive at inference time and how they handle hidden problem-state information before prediction. Some treat simulation as direct prediction from the visible field [1–3]; others reduce the difficulty with extra problem information or richer context, such as equations, coefficients, boundary conditions, or short trajectories [8–12]; and a related line explicitly recovers hidden state, parameters, operators, or memory before prediction [4, 5, 7, 13]. Amortized simulation-based inference (SBI) goes further and learns full posteriors over simulator parameters, often from learned summary statistics of the observation [14–16]. Yet single-observation simulation still lacks a principled reusable intermediate object. The missing problem state is usually left implicit, bypassed with extra information, or only partially recovered, and even a full posterior targets a parameter chosen in advance rather than the distinctions that downstream tasks require. As a result, ambiguity unresolved by one field is not preserved for downstream prediction or decision.

We address this gap with **posterior-first neural PDE simulation**:

$$x_t \longmapsto q_\phi\left(z_t \mid x_t\right) \longmapsto p_\psi\left(y \mid x_t, z_t\right).$$

Here x_t is the observed field at inference time, $q_\phi\left(z_t \mid x_t\right)$ is a posterior over latent problem states compatible with that field, and $p_\psi\left(y \mid x_t, z_t\right)$ denotes downstream prediction conditioned on the field and such a state. This factorization makes explicit what single-observation simulation must do under ambiguity: first infer which hidden problem states remain plausible from the visible field, then evolve or answer downstream queries conditional on that uncertainty. Its first stage is an amortized posterior estimator of the kind used in SBI, with fixed or learned summaries. What posterior-first adds is a task-to-target-to-interface construction: the declared tasks determine which hidden distinctions z_t must retain, and the inferred posterior stays operative in rollout. Concretely, in Section 3 we define the latent problem state and the posterior-first factorization and give a construction of this target from a declared task family. We then show in Section 4 that the Bayes-optimal reusable intermediate object is a posterior, that proper scoring rules identify this posterior as their population-risk minimizer, and that deterministic point latents face an explicit ambiguity barrier. To make this formulation concrete, we instantiate in Section 5 a minimal posterior-first simulator that infers a structured problem-state posterior and conditions evolution on it through a matched rollout backbone and a theorem-facing training objective.

Across the two empirical layers in Section 6, posterior-first models consistently outperform matched direct-rollout baselines, capacity-matched controls, and literature alternatives. Specifically, on the theorem-facing synthetic benchmark, they close most of the direct-to-oracle gap under controlled ambiguity. On the public PDEBench layer, they improve rollout error on every family, reducing the pooled public nRMSE from 0.175 to 0.132 while closing about 60% of the hidden-metadata oracle gap. Matched controls then separate estimator, target, and interface: task-aligned targeting and posterior-mixture deployment carry the gains under a fixed SBI estimator, posterior sampling beats auxiliary labels and MAP conditioning with a latent-diffusion backbone, a frozen posterior encoder transfers to held-out tasks with few labels, and the advantage persists in a compound stress test. These results support our core claim: single-observation neural PDE simulation should be posterior-first rather than a monolithic field-to-future map. In Section 7 we explain why this matters: making pre-rollout problem-state recovery explicit changes not only how neural simulators are built, but also what scientific surrogate models should be optimized and evaluated for.

2 Related Work

2.1 Types of Inputs to Neural Simulators

Research on neural PDE simulation can be classified by the inputs available to the simulator at inference time. A first class uses the observed field itself as the input and learns either a solution operator or a rollout map directly from fields or coordinates. FNO and DeepONet are canonical examples of this regime [1, 2]. PDE-Refiner is a representative long-horizon rollout model in the same setting [3]. A second class gives the simulator structured problem information in addition to the field. This information can include symbolic equations, coefficients, boundary conditions, geometry, or multimodal operator descriptions. Unisolver and PROSE exemplify this explicit-condition regime [8, 9]. A third class gives the simulator richer context before or during prediction. This context can take the form of short trajectories, contextual examples, or prompts from related dynamics. ICON, DISCO, and Zebra are representative methods in this class [10–12].

Taken together, these lines of work still leave single-observation simulation without a principled reusable intermediate object: direct-rollout methods leave the missing problem state implicit, whereas methods that avoid this difficulty do so by supplying extra information from outside the observation itself. Our work addresses this missing object directly with a posterior-first formulation of neural PDE simulation. This matters because it turns unresolved problem-state ambiguity into an explicit reusable simulator interface, rather than leaving it hidden inside end-to-end rollout or bypassing it with extra inputs, and thereby replaces an architecture-level modeling habit with a general principle for how scientific simulators should be built, reasoned about, and evaluated.

2.2 Problem-State Recovery Methods and What They Recover

Among methods that recover missing information before prediction, prior work falls into four broad groups: reconstructing hidden field state from partial observations [4, 13], identifying equations, parameters, coefficients, or source terms [5, 6], inferring operator or dynamics representations from richer context [10–12], and treating memory as part of the latent state [7]. Taken together, these methods remain partial: they recover only one slice of the hidden problem state or rely on richer context than a single observed field. As a result, ambiguity that is joint across hidden state, coefficients, operators, boundary conditions, and memory is still not exposed as a reusable object from one field. Our formulation instead targets a posterior over the joint task-sufficient problem state, so that unresolved ambiguity can be carried forward explicitly into downstream evolution and decision-making.

The closest machinery is amortized posterior estimation. Simulation-based inference (SBI) trains neural posterior estimators $q_\phi(\theta \mid s_\eta(x))$ with a fixed or jointly learned summary s_η of the observation [14–18], using spline flows, adversarial training, or conditional diffusion [19–21], and PDE pipelines infer a chosen parameter or function and may simulate from posterior samples [22, 23]. Supervised metadata conditioning feeds one predicted hidden label to the simulator, and latent diffusion emulators learn future-trajectory uncertainty once the simulation parameters are supplied [24]. These methods provide the estimator and the backbone, but none derives the target from the downstream task family or keeps its posterior operative in a neural rollout: SBI targets a parameter chosen in advance, hard conditioning collapses the posterior to a point, and a direct generator can match one task's future marginal without exposing which hidden problem produced it. Posterior-first simulation uses these components as they are and contributes this task-to-target-to-interface construction. This matters because finite simulation and model capacity are then spent on distinctions that can change the requested outputs, while the remaining ambiguity is carried forward as several plausible rollouts.

3 Posterior-First Problem-State Formulation

We formalize what to infer from one field, why the inferred object is a posterior, and how it is obtained from benchmark labels. The single-observation regime is hard because one observed field need not identify hidden state, coefficients, operators, boundary conditions, or memory relevant to downstream tasks. We therefore separate the observed field from the latent world that generated it: let W_t be the complete latent world state, and let $X_t = \mathcal{O}(W_t)$ be the visible field under observation map $\mathcal{O}$. We write x_t for a realized observation. Downstream scientific uses are indexed by $\tau \in \mathcal{T}$, with target variable Y_t^τ; the family $\mathcal{T}$ may include future fields, hidden-state queries, coefficient or operator queries, and uncertainty-sensitive decisions.

The reusable interface should not copy all of W_t. The full latent world is sufficient but may contain distinctions that no downstream task can ever use. We call $Z_t = g(W_t)$ *task-sufficient* if, once the observation X_t and Z_t are known, the remaining details of W_t do not change the law of any task target:

$$Y_t^\tau \perp W_t \mid (X_t, Z_t), \qquad \tau \in \mathcal{T}. \tag{1}$$

Sufficiency alone is still too weak, since many overly detailed states, including W_t itself, satisfy (1). We therefore identify, within each observation fiber, latent worlds that are indistinguishable to the whole task family. Let $\mathcal{L}(\cdot \mid \cdot)$ denote conditional law, and define

$$w \sim_{\mathcal{T}} w' \iff \mathcal{O}(w) = \mathcal{O}(w') \text{ and } \mathcal{L}(Y_t^\tau \mid W_t = w) = \mathcal{L}(Y_t^\tau \mid W_t = w') \text{ for all } \tau \in \mathcal{T}. \tag{2}$$

Let $Z_t^\star = g^\star(W_t)$ be the quotient label induced by this relation. It is the minimal task-sufficient problem state: given the observed field, it keeps exactly the latent distinctions that can change at least

one downstream law and discards the rest. Appendix A.1 gives the formal quotient construction and proves that every other task-sufficient state is a refinement of $Z_t^\star$.

Even the minimal state is not necessarily identified by one observed field. Multiple values of $Z_t^\star$ can remain compatible with the same x_t. The object to infer is therefore not a point latent, but the posterior over task-sufficient problem states:

$$\pi_t(\cdot \mid x_t) := \mathbb{P}(Z_t^\star \in \cdot \mid X_t = x_t)\,. \tag{3}$$

While $Z_t^\star$ is minimal for the whole family, a single task depends only on a coarser projection of it. For a task τ, let $z_\tau = g_\tau(z)$ be this task-relevant projection of z, which merges states that induce the same law of Y_t^τ, and let $\pi_{t,\tau}(\cdot \mid x_t) := (g_\tau)_\# \pi_t(\cdot \mid x_t)$. With x_t and τ held fixed, prediction integrates only the uncertainty in this projection:

$$p_\tau(y \mid x_t) = \int p_\tau(y \mid x_t, z_\tau)\, \pi_{t,\tau}(dz_\tau \mid x_t), \tag{4}$$

where $p_\tau(y \mid x_t, z_\tau)$ denotes the predictive law of Y_t^τ given the field and the projected state, written in density notation. Integrating over all of $Z_t^\star$ gives the same law by construction of g_τ. Thus the visible field is first converted into uncertainty over plausible problem states, and downstream prediction is then conditional on that uncertainty. Appendix A.2 gives the measure-theoretic derivation.

In benchmarks, $Z_t^\star$ is the theorem-level object but is not directly annotated. We instead observe a countable semantic refinement C_t, with $Z_t^\star = h(C_t)$, whose labels record the hidden problem-state slices available for supervision. If $r_t(c \mid x_t) = \mathbb{P}(C_t = c \mid X_t = x_t)$ is the refinement posterior, then the desired task-sufficient posterior is its pushforward:

$$\pi_t(z \mid x_t) = \sum_{c:\, h(c)=z} r_t(c \mid x_t). \tag{5}$$

Appendix A.3 formalizes this refinement bridge. Hence the core inference target is $\pi_t(\cdot \mid x_t)$, represented in experiments by a supervised refinement posterior.

The map h is fixed by design rather than learned, and it must preserve sufficiency. On a finite catalogue with common support, let $K_\tau(\cdot \mid x, c) := \mathbb{P}(Y_t^\tau \in \cdot \mid X_t = x, C_t = c)$, identify $c \sim_{\mathcal{T}} c'$ when these kernels agree for every declared task and almost every x, and set $h(c) := [c]_{\mathcal{T}}$. This h satisfies $Y_t^\tau \perp C_t \mid (X_t, h(C_t))$ and is the coarsest map that does so (Appendix A.3). Operationally, one declares the task family and uses the governing equations or simulator dependencies, together with the task functionals, to keep every catalogue factor that can change some task kernel, merging only distinctions certified to be task-invariant; keeping an uncertified distinction preserves sufficiency at the price of an over-refined target. In our Shallow-Water protocol, for example, x_t is the initial water-height field and C_t indexes eight joint source/boundary states; a boundary query depends only on the boundary coordinate, whereas rollout may depend on both. Single-observation neural PDE simulation should be posterior-first: problem-state posterior inference followed by conditional evolution, rather than a monolithic field-to-future map.

4 Why the Reusable Intermediate Object Is a Posterior

Section 3 defined the posterior over the minimal task-sufficient problem state. We now argue that it is the right object between observation and evolution because it is sufficient, learnable, irreplaceable by a point, and worth learning well: Bayes decisions depend on the observation only through it (Section 4.1); proper scoring rules on refinement labels identify it, whereas any point latent remains an ambiguity barrier away from it (Section 4.2); and its error bounds the downstream utility gap, while rollout-only training does not identify it (Section 4.3). Appendix B contains the full derivations.

4.1 The Correct Reusable Intermediate Object Is a Task-Sufficient Posterior

Fix a task $\tau \in \mathcal{T}$, action space $\mathcal{A}_\tau$, action $a \in \mathcal{A}_\tau$, and task loss $\ell_\tau(a, Y_t^\tau)$. Let $\rho_\tau(a, z) := \mathbb{E}[\ell_\tau(a, Y_t^\tau) \mid Z_t^\star = z]$. For a realized observation x_t, the conditional Bayes value satisfies

$$\begin{aligned} \mathcal{V}_\tau(a, x_t) &:= \mathbb{E}[\ell_\tau(a, Y_t^\tau) \mid X_t = x_t] \\ &= \mathbb{E}[\rho_\tau(a, Z_t^\star) \mid X_t = x_t] \\ &= \int \rho_\tau(a, z)\, \pi_t(dz \mid x_t). \end{aligned} \tag{6}$$

Task-sufficiency gives the second line, and conditioning on $\pi_t(\cdot \mid x_t)$ gives the third. Hence Bayes actions depend on x_t only through this posterior, making the reusable object posterior-valued rather than point-valued. For the benchmark refinement, the same value is obtained from r_t via the pushforward in (5).

4.2 The Posterior Is Learnable and Point Latents Are Ambiguity-Limited

Let $\mathcal{C} = \mathrm{range}(C_t)$, let $r_X(\cdot) := r_t(\cdot \mid X_t)$ be the true refinement posterior, and let $q_X(\cdot)$ be a predicted distribution on $\mathcal{C}$. For the log score $\mathcal{L}_{\log}(q) := \mathbb{E}[-\log q_X(C_t)]$, the excess risk over the Bayes risk $\mathcal{L}^\star_{\log}$ is

$$\mathcal{L}_{\log}(q) - \mathcal{L}^\star_{\log} = \mathbb{E}\left[\sum_{c\in\mathcal{C}} r_X(c) \log \frac{r_X(c)}{q_X(c)}\right] = \mathbb{E}[\mathrm{KL}(r_X \,\|\, q_X)]\,. \tag{7}$$

The log score therefore already identifies the refinement posterior: for a well-specified model class, its unique population-risk minimizer is $q_X = r_X$. The multiclass Brier score $\mathcal{L}_{\mathrm{Br}}(q) := \mathbb{E}[\|q_X - e_{C_t}\|_2^2]$ plays a different role; let e_c be the one-hot distribution at label c. Since $\mathbb{E}[e_{C_t} \mid X_t] = r_X$, conditioning on X_t gives the orthogonal decomposition

$$\mathbb{E}\big[\|q_X - e_{C_t}\|_2^2 \mid X_t\big] = \|q_X - r_X\|_2^2 + \mathbb{E}\big[\|r_X - e_{C_t}\|_2^2 \mid X_t\big]\,.$$

Thus, with $\mathcal{L}^\star_{\mathrm{Br}}$ denoting the Bayes Brier risk,

$$\mathcal{L}_{\mathrm{Br}}(q) - \mathcal{L}^\star_{\mathrm{Br}} = \mathbb{E}\big[\|q_X - r_X\|_2^2\big]\,. \tag{8}$$

This exact squared-L_2 regret is not needed for identification; it supplies the geometry of point collapse below and the finite-$\mathcal{C}$ utility bound in Section 4.3.

The same calculation explains why deterministic point latents are insufficient in genuinely ambiguous observations. A point predictor $\hat{c}(X_t)$ represents a one-hot distribution. For any posterior r, $\|e_c - r\|_2^2 = 1 - 2r(c) + \|r\|_2^2$, so the best one-hot approximation incurs

$$\Gamma_{\mathrm{amb}}(r) := \min_c \|e_c - r\|_2^2 = 1 + \|r\|_2^2 - 2\|r\|_\infty \geq (1 - \|r\|_\infty)^2. \tag{9}$$

This ambiguity penalty vanishes only for Dirac posteriors. Therefore, when one field leaves multiple problem states plausible, a point latent is separated from the posterior target by an irreducible statistical gap.

4.3 Posterior Quality Controls Simulator Utility

Posterior recovery matters because downstream value is linear in the posterior. For an approximate refinement posterior q_{x_t}, define $\mathcal{V}^q_\tau(a, x_t) := \sum_{c\in\mathcal{C}} \rho_\tau(a, h(c)) q_{x_t}(c)$. If $B_\tau := \sup_{a\in\mathcal{A}_\tau, c\in\mathcal{C}} |\rho_\tau(a, h(c))| < \infty$, then for every action a,

$$\big|\mathcal{V}^q_\tau(a, x_t) - \mathcal{V}_\tau(a, x_t)\big| \leq B_\tau \, \|q_{x_t} - r_t(\cdot \mid x_t)\|_1. \tag{10}$$

Let $a^q_\tau(x_t)$ minimize the approximate value $\mathcal{V}^q_\tau$, and let $a^\star_\tau(x_t)$ minimize the true value $\mathcal{V}_\tau$. Then

$$\mathcal{V}_\tau\big(a^q_\tau(x_t), x_t\big) - \mathcal{V}_\tau\big(a^\star_\tau(x_t), x_t\big) \leq 2B_\tau \, \|q_{x_t} - r_t(\cdot \mid x_t)\|_1. \tag{11}$$

Proper scoring losses then translate into utility control. By Pinsker's inequality and (7), and, if $\mathcal{C}$ is finite, by $\|v\|_1 \leq \sqrt{|\mathcal{C}|}\|v\|_2$ and (8),

$$\mathbb{E}[\|q_X - r_X\|_1] \leq \sqrt{2\big(\mathcal{L}_{\log}(q) - \mathcal{L}^\star_{\log}\big)}, \qquad \mathbb{E}[\|q_X - r_X\|_1] \leq \sqrt{|\mathcal{C}|\big(\mathcal{L}_{\mathrm{Br}}(q) - \mathcal{L}^\star_{\mathrm{Br}}\big)}. \tag{12}$$

Thus posterior metrics are not merely auxiliary diagnostics: they upper-bound the downstream utility gap of the simulator induced by the learned posterior.

Finally, rollout-only hidden states are not generally identified as reusable objects. Suppose training uses only a restricted rollout family $\mathcal{T}_0 \subsetneq \mathcal{T}$. If two distinct posteriors $\pi \neq \pi'$ over $Z^\star_t$ satisfy $\int \rho_\tau(a, z)\, \pi(dz) = \int \rho_\tau(a, z)\, \pi'(dz)$ for all $\tau \in \mathcal{T}_0$ and $a \in \mathcal{A}_\tau$, then every objective built only from $\mathcal{T}_0$ is indifferent between them, even though an omitted task in $\mathcal{T}$ may distinguish them. A rollout hidden state can therefore be sufficient for the training rollout loss while failing as a reusable interface. Explicit posterior recovery avoids this non-identification by targeting the task-sufficient posterior itself.

5 A Minimal Posterior-First Simulator

We instantiate the theory with the smallest architectural change that makes the inferred posterior explicit. In experiments, $z_t \in \mathcal{Z}$ is the realized label of the finite annotated semantic refinement C_t used for supervision: its coordinates cover the hidden field state, coefficient or operator regime, boundary class, or memory class, while $Z_t^\star$ remains the theorem-level quotient and x_t remains the visible rollout input. Given the observed field x_t, a posterior encoder predicts

$$q_\phi(z \mid x_t) \approx \mathbb{P}(C_t = z \mid X_t = x_t), \qquad z \in \mathcal{Z}. \tag{13}$$

The output is a full joint posterior. This countable annotated refinement is the smallest setting in which the proper-scoring identities of Section 4 can be tested directly. The encoder is a seven-layer CNN (widths 32 to 256, GroupNorm and SiLU) with a softmax head over the 16/16/8/16 catalogue states of DR/DS/SW/INS or the 64 synthetic states; continuous metadata cells are fitted on the training split and then frozen.

The rollout still uses x_t as the visible initial condition; the posterior supplies the unresolved problem-state information. Ideally, prediction follows the finite mixture

$$\sum_{z \in \mathcal{Z}} q_\phi(z \mid x_t) p_\psi(y \mid x_t, z),$$

where p_ψ is a conditional rollout predictor. In the main experiments, we keep one matched rollout backbone and expose the posterior through the embedding summary

$$s_\phi(x_t) = \sum_{z \in \mathcal{Z}} q_\phi(z \mid x_t) e(z), \qquad \hat{y} = F_\psi(x_t, s_\phi(x_t)). \tag{14}$$

Here $e(z) = E_{z,:}$ is a row of a trainable codebook $E \in \mathbb{R}^{|\mathcal{Z}| \times 64}$, so $s_\phi(x_t) = q_\phi(\cdot \mid x_t)^\top E$ averages, in the Shallow-Water example, the codes of the eight source/boundary states by their posterior weights; a linear projection adds it to the features after each backbone's input embedding (FNO lifting layer, U-Net stem, ConvLSTM encoder, or Transformer patch embedding), it is computed once from x_t and reused at every rollout step, and F_ψ has the same rollout family as the direct baseline. Because nonlinear F_ψ need not commute with posterior averaging, (14) is a tested low-cost approximation to the mixture; Appendix C.2 compares it with explicit mixtures and posterior sampling, which we use when the full predictive law is needed (Section 6.2.3).

Training is a single end-to-end optimization that learns the posterior with a proper scoring rule and trains the simulator with the same downstream losses used by the direct baselines:

$$\min_{\phi,\psi} \ \lambda_{\text{post}} \mathbb{E}[S(q_\phi(\cdot \mid x_t), z_t)] + \sum_{\tau \in \mathcal{T}_{\text{train}}} \lambda_\tau \mathbb{E}[\ell_\tau(F_\psi(x_t, s_\phi(x_t)), Y_t^\tau)]\,. \tag{15}$$

Here S is the log score on z_t, the task losses are those of the direct baselines (normalized rollout MSE on the public benchmark), and the posterior and total task terms have unit weights. The log score trains the encoder directly, and the task losses backpropagate through s_ϕ into the encoder, codebook, projection, and simulator without stop-gradient. The theory guarantees reuse of the object (x_t, π_t); because the task losses also shape q_ϕ and E, reuse of the trained encoder is tested empirically (Section 6.2.3), while F_ψ is task-specific. Implementation details and summary-fidelity checks are in Appendix C.

The simulator is minimal for the empirical test: posterior-first and direct models share the rollout family, optimization budget, and task supervision. Removing $s_\phi(x_t)$ gives the field-to-future baseline, and replacing $q_\phi(\cdot \mid x_t)$ by a point estimate gives the point-latent control. The comparison is designed to isolate latent-state posterior recovery.

6 Empirical Evaluation

Our experiments test three claims. First, when a single field is ambiguous, posterior-valued problem-state inference should outperform point-valued or direct interfaces and approach the oracle state interface. Second, when hidden problem metadata are withheld in public PDEBench tasks, the same posterior-first mechanism should improve standard rollout under matched backbones [25]. Third, these gains should come from the task-aligned target and the posterior interface rather than from the posterior estimator, hidden-state labels, or the rollout backbone.

6.1 Experimental setup

6.1.1 Datasets and scope

We use two benchmark layers. The synthetic layer is an in-house exact-collision generator built from Gray–Scott-style reaction–diffusion rendering [26, 27]. Each example contains one scalar field $x_t \in \mathbb{R}^{1\times 64\times 64}$, a discrete semantic state $z_t \in \mathcal{Z}_{\text{syn}}$, the exact posterior $r_X(\cdot) = \mathbb{P}(z_t \in \cdot \mid X_t)$, and task targets for future fields, operator class, coefficient value, and uncertainty. We use $|\mathcal{Z}_{\text{syn}}| = 64$ states and four ambiguity regimes—near-Dirac, mid, high, and very high—formed by increasing the number of compatible semantic states in each collision class. Full generator rules and splits are given in Appendix D.2.

The public layer is a metadata-hidden protocol built on PDEBench [25] with four forward families, Diffusion-Reaction (DR), Diffusion-Sorption (DS), Shallow-Water (SW), and Incompressible Navier–Stokes (INS), each rolled out for 100 steps from one initial field. Each item exposes only this field x_t, while selected metadata are withheld and used as z_t^{pub}: regime/coefficient in DR, coefficients in DS, joint source and boundary class in SW, and forcing or viscosity in INS. Direct baselines use only x_t, posterior-first models infer $q(z_t^{\text{pub}} \mid x_t)$, and oracle controls receive the metadata. Grids, sequence counts, taxonomies, splits, and probe definitions are reported in Appendix D.4.

6.1.2 Compared methods, backbones, and regimes

We organize the compared methods by the alternative explanation they test. In the synthetic layer, where the true posterior and oracle state are known, we use the full matched control ladder. `Direct-Point` is the standard deterministic field-to-future rollout model. `Direct-Dist` keeps the same direct interface but predicts a heteroscedastic output distribution, testing output uncertainty alone. `Direct-MTL` adds auxiliary hidden-state supervision to the direct model, testing latent supervision without a posterior interface. `Direct-Large` widens and deepens the direct model, testing capacity. `Point-`z first predicts a single semantic state $\hat{z}_t$ and then conditions rollout on it, while `Posterior-`z predicts the full $q(z_t \mid x_t)$ before rollout. `RichContext-Posterior-`z receives a 4-frame context that resolves most collisions, and `Oracle-`z receives the true latent problem state; these two play the role of the richer-context and explicit-condition classes of Section 2 and serve as upper references, because they receive information that a single field does not provide. Architectures, losses, and configurations are given in Appendix C and the released code.

In the public PDEBench layer, the main text keeps the comparison smaller and focused on external validity: `Direct-Point`, `Direct-Dist`, `Posterior-`z, and `Oracle-`z, asking whether explicit hidden-metadata posterior recovery closes the direct-to-oracle rollout gap under matched conditions. Appendices D.5 and D.6 add external probabilistic baselines where their native assumptions fit the metadata-hidden task, namely Approximate Bayesian Neural Operators (ABNO) [28], the invertible neural-operator posterior of Kaltenbach et al. [29], and the label-free `DeepEns-Direct` [30] and `GateMoE-Direct` [31], to test whether generic probabilistic direct modeling can replace explicit problem-state posterior recovery.

For rollout backbones, we use four common neural-simulator families with complementary inductive biases: U-Net [32], FNO [1], ConvLSTM [33], and Transformer [34]. Within each backbone family, all matched internal controls share the same rollout stem, optimization budget, train/validation/test split, and observation protocol. The public experiments pool over the four families and all matched backbones; external probabilistic baselines are evaluated family-wise where their assumptions apply.

6.1.3 Metrics and evaluation protocol

For the synthetic benchmark, each method is trained on the synthetic training split, selected by validation task-bundle risk, and evaluated on held-out test examples from the same ambiguity regimes. The mechanism metric is the average ambiguity barrier $\overline{\Gamma}_{\text{amb}}$, the test mean of $\Gamma_{\text{amb}}(r_i)$ in (9) under the exact posterior r_i; it is the best Brier error that any deterministic point latent can achieve, and we compare it with the observed *Point gap*, the excess latent-state Brier error of `Point-`z over `Posterior-`z. Downstream, we report the remaining direct-to-oracle gap $G_{\text{rem}}(m) := (R_{\text{full}}(m) - R_{\text{full}}^{\text{ora}})/(R_{\text{full}}^{\text{dir}} - R_{\text{full}}^{\text{ora}})$ of the full task-bundle risk, which is 1 for the direct baseline and 0 for the oracle, and the negative log-likelihood of the 8-step future field (Future NLL).

Table 1: Synthetic exact-ambiguity mechanism check across ambiguity regimes and identifiability controls.

Regime / control	$\mathbb{E}[\Gamma_{\mathrm{amb}}(r_X)]$	**Point gap**	**gap /** $\mathbb{E}[\Gamma_{\mathrm{amb}}]$
Near-Dirac control	0.004	0.004	1.00
Mid ambiguity	0.018	0.016	0.89
High ambiguity	0.031	0.028	0.90
Very high ambiguity	0.047	0.042	0.89
4-frame richer-context control	0.005	0.005	1.00
Operator-token Dirac control	0.003	0.003	1.00

Table 2: Synthetic downstream performance on the full task bundle.

Method	**Future NLL**	G_{rem}
`Direct-Point`	1.48	1.000
`Direct-Dist`	1.22	0.732
`Direct-MTL`	1.24	0.498
`Direct-Large`	1.20	0.598
`Point-`z	1.18	0.460
`Posterior-`z	1.02	0.381
`RichContext-Posterior-`z	0.92	0.205
`Oracle-`z	0.81	0.000

For the public PDEBench benchmark, each method is evaluated by closed-loop rollout on held-out trajectories with the trajectory-wise relative error nRMSE and the oracle-gap closure $\mathrm{GapClose}(m) := (\mathrm{nRMSE}^{\mathrm{dir}} - \mathrm{nRMSE}(m))/(\mathrm{nRMSE}^{\mathrm{dir}} - \mathrm{nRMSE}^{\mathrm{ora}})$, which is 0 without improvement over direct prediction and 1 when the oracle gap is fully closed. For stratified analysis, we compute GapClose within four bins of a held-out single-field ambiguity score $a(x_t)$, with the high-ambiguity value taken from the highest-score bin. To check whether rollout gains coincide with hidden problem-state recovery, we report the strongest metadata probe for each family: accuracy for discrete metadata and R^2 for continuous metadata, averaged across families in the pooled row. The controls in Sections 6.2.3–6.2.4 also use the CRPS of a scalar parameter posterior and the energy score of sampled rollouts, two proper scores for which lower is better. Results are averaged over five training seeds; full definitions are given in Appendix D.1.

6.2 Main results

6.2.1 Synthetic exact-ambiguity benchmark: mechanism and downstream closure

The synthetic benchmark is the only setting in which the exact posterior and the ambiguity barrier are known. Table 1 therefore reports the mechanism check predicted by Section 4.1, and Table 2 reports the corresponding downstream consequence on the full task bundle.

Table 1 shows that the deterministic point-versus-posterior gap is governed by the ambiguity barrier rather than by generic model capacity. As ambiguity increases from the mid to the very-high regime, the expected barrier rises from 0.018 to 0.047 and the observed point gap rises in step from 0.016 to 0.042, while the ratio between them stays concentrated at 0.89–0.90. In the near-Dirac, 4-frame, and operator-token controls, both the barrier and the observed gap collapse to 0.003–0.005. The synthetic mechanism result therefore matches the theorem in both directions: the point-latent penalty appears when a single field is ambiguous, and it disappears again when the observation becomes identifying.

Table 2 shows how that mechanism translates into simulator utility. Relative to `Direct-Point`, a larger direct model and auxiliary hidden-target supervision reduce the remaining direct-to-oracle gap to 0.598 and 0.498, respectively, but neither matches an explicit posterior stage. The decisive comparison is between `Point-`z and `Posterior-`z: replacing a point latent by a posterior lowers future NLL from 1.18 to 1.02 and reduces G_{rem} from 0.460 to 0.381. The richer-context and oracle controls give the expected ceilings at 0.205 and 0.000. Appendix D.3 shows that the same ordering persists on the fuller task bundle and across ambiguity regimes (Tables 6–7), and its rollout-only ablation (Table 8) shows why rollout accuracy cannot identify the reusable state: across five training objectives the training-task nRMSE stays at 0.028–0.030, while the hidden-task score rises from 0.61 for rollout-only training to 0.84 with explicit posterior recovery over the full task family. Appendix C.2 then checks the implementation approximation directly: replacing the summary interface by an explicit mixture changes synthetic G_{rem} only from 0.381 to 0.365, while a four-sample approximation gives 0.373.

Taken together, the synthetic layer supports our theorem-facing claim: when ambiguity can be controlled exactly, explicit posterior inference is the mechanism that explains the downstream gain.

Table 3: Public PDEBench rollout under the metadata-hidden protocol: nRMSE of each method (lower is better) and GapClose of `Posterior-`z (higher is better); High-amb. uses each family's highest ambiguity bin.

Family	**Direct**	**Direct-Dist**	**Posterior-**z	**Oracle-**z	GapClose	**High-amb.** GapClose
DR	0.2228	0.1957	0.1690	0.1290	0.573	0.669
DS	0.0285	0.0222	0.0182	0.0125	0.641	0.740
SW	0.0772	0.0607	0.0532	0.0387	0.623	0.722
INS	0.3702	0.3311	0.2872	0.2305	0.594	0.688
All	0.1747	0.1524	0.1319	0.1027	0.594	0.689

6.2.2 Public PDEBench metadata-hidden oracle-gap probe

Because the public benchmark has hidden metadata but no exact posterior labels, this subsection uses three complementary views: gap closure as a function of ambiguity (Figure 1), rollout error per family (Table 3), and recovery of the hidden metadata (Table 12). Figure 1 bins test samples, pooled across families, by a held-out single-field ambiguity score $a(x_t)$, which estimates how much oracle benefit remains under a single visible field (Appendix D.4). As $a(x_t)$ rises from 0.14 to 0.89 across bins, `Posterior-`z closes more of the direct-to-oracle gap, increasing from 0.385 to 0.744. This gives the ambiguity-stage conclusion: posterior-first simulation helps most on public samples where more hidden problem state remains unresolved.

Table 3 reports the rollout errors and direct-to-oracle gap closure under the metadata-hidden protocol. `Posterior-`z lowers nRMSE in all four public families. In the `All` row, averaged over the public evaluation settings, nRMSE drops from 0.1747 to 0.1319, compared with 0.1524 for `Direct-Dist`, which models output uncertainty alone, and the oracle reference 0.1027. This corresponds to 59.4% direct-to-oracle gap closure overall and 68.9% on each family's highest-ambiguity bin, whereas the 0.744 of Figure 1 comes from bins pooled across families. Family-wise closures are all positive, ranging from 0.573 to 0.641, so the gain is not driven by one PDE family. Appendix D.5 shows the same ordering for every backbone and against external probabilistic baselines. This gives the rollout-stage conclusion: the ambiguity-sensitive mechanism translates into robust prediction improvement.

Table 12 in Appendix D.5 reports the strongest available hidden-metadata probe for each public family. In the `All` row, the mean probe score rises from 0.628 for the direct baseline to 0.800 for `Posterior-`z, approaching the oracle upper bound of 0.901. The improvement appears in every family, for both discrete metadata accuracy and continuous-metadata R^2. That appendix also reports calibration diagnostics, and Appendix C.2 checks the summary-interface approximation. This gives the hidden-state conclusion: the rollout gain comes with better recovery of withheld problem-state information, not only better marginal predictive uncertainty.

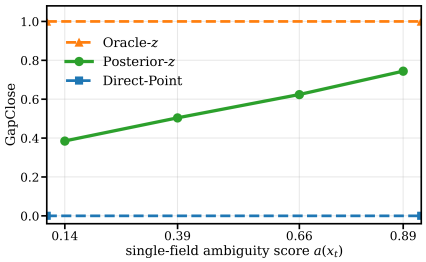


Figure 1: GapClose versus held-out single-field ambiguity score $a(x_t)$ on metadata-hidden PDEBench. Points show `Posterior-`z bin means; dashed lines mark `Direct-Point` at 0 and `Oracle-`z at 1.

Taken together, the public layer supports the practical half of the claim: posterior-first simulation improves standard PDEBench rollout by recovering hidden problem-state information that direct single-field predictors leave implicit.

6.2.3 Estimator, target, and interface controls

Because any amortized posterior estimator can implement the posterior stage, we test whether the gains come from the estimator or from the target and interface (Appendix D.7 gives protocols and five-seed standard errors). In Test A, on PDEArena's Navier–Stokes task with hidden buoyancy [35], a learned CNN summary improves posterior CRPS and final-quarter rollout Energy over dimension-matched PCA (from 0.2116/0.1517 to 0.1916/0.1459), whereas a spline-flow NPE [19] and a conditional-diffusion NPE [21] on the same summary are comparable. In Test B, the synthetic benchmark gains a task-irrelevant eight-state nuisance U, so $C = (Z^\star, U)$ and $h(z, u) = z$. With

the NPE family fixed, inferring $Z^\star$ directly instead of C followed by exact pushforward lowers the posterior error from 0.0294 to 0.0187 and the simulations needed to reach 0.025 from 79.4k to 30.4k, an operator-only slice misses this threshold even at 160k, and deploying the same posterior as a mixture instead of its MAP state lowers all-task risk from 0.3391 to 0.3196. A learned-summary SBI pipeline thus instantiates posterior-first simulation once it is given the task-aligned target and the posterior-mixture interface, and these two choices, not the estimator, carry the gains.

The same holds for label access and the rollout backbone. With the latent-diffusion emulator of Rozet et al. [24] on the metadata-hidden PDEBench splits, posterior sampling closes 59.0% of the nRMSE gap (from 0.1744 to 0.1323, oracle 0.1030), reproducing the 59.4% of the main backbones, whereas the same labels used only as an auxiliary target close about 4%. With one shared frozen q and checkpoint, sampling beats the MAP state in Energy by 0.0052 in the lowest-ambiguity stratum and by 0.0435 in the highest, and the exposed posterior answers a Shallow-Water boundary query better than the auxiliary head (Brier 0.1169 vs. 0.1317). Because the task losses also update the encoder, a four-fold leave-one-task-out test checks its reuse: a frozen encoder trained only with the proper score, with a head trained on 10% of the held-out task's labels, beats its own MAP state (risk 0.3618 vs. 0.3954) and retains 96.4% of the gap closure of an encoder trained on the other three tasks, supporting frozen-encoder, few-shot reuse within one supervised ontology.

6.2.4 Compound stress test

Finally, one nested compound test on the hidden-buoyancy task of Test A replaces the discrete head by a conditional spline flow, masks the initial field following POBench-PDE [36] and adds noise, shifts buoyancy out of its training range, and rolls out for 200 unseen steps (Appendix D.7). Propagating 64 posterior samples instead of the posterior mean of the same frozen flow lowers final-quarter Energy from 0.1493 to 0.1438 on clean in-distribution fields, from 0.2036 to 0.1742 with sparse noisy fields, and from 0.2476 to 0.2058 once parameter shift is added, even as the flow's 90% coverage falls from 90.4% to 83.6%. The advantage thus persists and grows as sensing degrades and then parameters shift.

7 Discussion

This paper argued that single-observation neural PDE simulation should be posterior-first rather than a monolithic field-to-future map. The contribution is not another estimator, backbone, or simulator head, but a change in the interface between observation and evolution, together with its theory: the declared tasks determine which hidden distinctions to infer, and before rollout the model should recover the distribution over the plausible problem states left unresolved by the observed field and keep it operative in evolution. This reframes the design question from building stronger hidden states inside end-to-end predictors to inferring the latent problem state needed for evolution. It also reframes evaluation: average rollout accuracy is not enough if the simulator is wrong about the hidden physics. Posterior-first simulation makes problem-state recovery, ambiguity sensitivity, and hidden-physics correctness first-class criteria alongside rollout error. In feedback-based active flow control [37], for example, a learned simulator could carry its posterior over unresolved forcing or boundary conditions into every control decision instead of committing to a single guess.

Our claim is bounded to single-observation simulation with a declared task family, where latent ambiguity is intrinsic and posterior recovery is most consequential. First, the theory assumes a fixed map h; we construct it when task kernels can be certified, but our experiments fix the catalogue from the known data-generating mechanism, and discovering a minimal quotient from data remains open. Second, posterior learning uses hidden-state labels; weakly supervised or unsupervised recovery was not tested, and performance degrades as labels become scarce (Appendix D.6). Third, the main simulator consumes a posterior summary that is close to explicit mixtures in our settings (Appendix C.2) but not lossless. Fourth, evidence beyond the discrete clean setting comes from one compound test that does not isolate each stressor, and reuse is shown only for a frozen encoder with few labels within one supervised ontology. Finally, as in Section 4, the posterior-to-utility bridge depends on task-family and stability assumptions, and closed-loop control was not validated.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (Grant No. 62372409), the Self-funding of the State Key Laboratory of Ocean Sensing (No. OSZ-2025PY19) and the Ministry of Science and Technology of the People's Republic of China (Grant No. 2023ZD0120704 of Project No. 2023ZD0120700). The authors declare no competing interests.

References

[1] Li, Z., Kovachki, N. B., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A. M., and Anandkumar, A. (2021). Fourier Neural Operator for Parametric Partial Differential Equations. In *International Conference on Learning Representations*.

[2] Lu, L., Jin, P., Pang, G., Zhang, Z., and Karniadakis, G. E. (2021). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3, 218–229.

[3] Lippe, P., Veeling, B., Perdikaris, P., Turner, R., and Brandstetter, J. (2023). PDE-Refiner: Achieving Accurate Long Rollouts with Neural PDE Solvers. *Advances in Neural Information Processing Systems*, 36, 67398–67433.

[4] Raissi, M., Yazdani, A., and Karniadakis, G. E. (2020). Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481), 1026–1030.

[5] Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). 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.

[6] Raissi, M. (2018). Deep Hidden Physics Models: Deep Learning of Nonlinear Partial Differential Equations. *Journal of Machine Learning Research*, 19(25), 1–24.

[7] Buitrago Ruiz, R., Marwah, T., Gu, A., and Risteski, A. (2025). On the Benefits of Memory for Modeling Time-Dependent PDEs. In *International Conference on Learning Representations*.

[8] Zhou, H., et al. (2025). Unisolver: PDE-Conditional Transformers Towards Universal Neural PDE Solvers. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 79061–79088.

[9] Liu, Y., Zhang, Z., and Schaeffer, H. (2024). PROSE: Predicting Multiple Operators and Symbolic Expressions using multimodal transformers. *Neural Networks*, 180, 106707.

[10] Yang, L., et al. (2023). In-context operator learning with data prompts for differential equation problems. *Proceedings of the National Academy of Sciences*, 120(39), e2310142120.

[11] Morel, R., Han, J., and Oyallon, E. (2025). DISCO: learning to DISCover an evolution Operator for multi-physics-agnostic prediction. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 44750–44774.

[12] Serrano, L., et al. (2025). Zebra: In-Context Generative Pretraining for Solving Parametric PDEs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 53940–53988.

[13] Williams, J. P., Zahn, O., and Kutz, J. N. (2024). Sensing with shallow recurrent decoder networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 480(2298), 20240054.

[14] Cranmer, K., Brehmer, J., and Louppe, G. (2020). The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48), 30055–30062.

[15] Radev, S. T., Mertens, U. K., Voss, A., Ardizzone, L., and Köthe, U. (2022). BayesFlow: Learning Complex Stochastic Models With Invertible Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4), 1452–1466.

[16] Chen, Y., Zhang, D., Gutmann, M. U., Courville, A., and Zhu, Z. (2021). Neural Approximate Sufficient Statistics for Implicit Models. In *International Conference on Learning Representations*.

[17] Papamakarios, G. and Murray, I. (2016). Fast ε-free Inference of Simulation Models with Bayesian Conditional Density Estimation. In *Advances in Neural Information Processing Systems*, 29.

[18] Tejero-Cantero, A., Boelts, J., Deistler, M., Lueckmann, J.-M., Durkan, C., Gonçalves, P. J., Greenberg, D. S., and Macke, J. H. (2020). sbi: A toolkit for simulation-based inference. *Journal of Open Source Software*, 5(52), 2505.

[19] Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. (2019). Neural Spline Flows. In *Advances in Neural Information Processing Systems*, 32.

[20] Ramesh, P., Lueckmann, J.-M., Boelts, J., Tejero-Cantero, Á., Greenberg, D. S., Gonçalves, P. J., and Macke, J. H. (2022). GATSBI: Generative Adversarial Training for Simulation-Based Inference. In *International Conference on Learning Representations*.

[21] Chen, T., Bansal, V., and Scott, J. G. (2025). Conditional Diffusions for Amortized Neural Posterior Estimation. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 2377–2385.

[22] Hull, R., Leonarduzzi, E., De La Fuente, L., Viet Tran, H., Bennett, A., Melchior, P., Maxwell, R. M., and Condon, L. E. (2024). Simulation-based inference for parameter estimation of complex watershed simulators. *Hydrology and Earth System Sciences*, 28(20), 4685–4713.

[23] Moss, G., Muhle, L., Drews, R., Macke, J. H., and Schröder, C. (2025). FNOPE: Simulation-based inference on function spaces with Fourier Neural Operators. In *Advances in Neural Information Processing Systems*, 38.

[24] Rozet, F., Ohana, R., McCabe, M., Louppe, G., Lanusse, F., and Ho, S. (2025). Lost in Latent Space: An Empirical Study of Latent Diffusion Models for Physics Emulation. In *Advances in Neural Information Processing Systems*, 38.

[25] Takamoto, M., Praditia, T., Leiteritz, R., MacKinlay, D., Alesiani, F., Pflüger, D., and Niepert, M. (2022). PDEBench: An Extensive Benchmark for Scientific Machine Learning. In *36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks*.

[26] Gray, P. and Scott, S. K. (1984). Autocatalytic reactions in the isothermal, continuous stirred tank reactor: Oscillations and instabilities in the system $A + 2B \rightarrow 3B$, $B \rightarrow C$. *Chemical Engineering Science*, 39(6), 1087–1097.

[27] Pearson, J. E. (1993). Complex patterns in a simple system. *Science*, 261(5118), 189–192.

[28] Magnani, E., Krämer, N., Eschenhagen, R., Rosasco, L., and Hennig, P. (2025). Approximate Bayesian Neural Operators: Uncertainty Quantification for Parametric PDEs. *Transactions on Machine Learning Research*.

[29] Kaltenbach, S., Perdikaris, P., and Koutsourelakis, P.-S. (2023). Semi-supervised invertible neural operators for Bayesian inverse problems. *Computational Mechanics*, 72(3), 451–470.

[30] Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles. In *Advances in Neural Information Processing Systems*, 30.

[31] Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. (2017). Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *International Conference on Learning Representations*.

[32] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer.

[33] Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-C. (2015). Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Advances in Neural Information Processing Systems*, 28, 802–810.

[34] Vaswani, A., et al. (2017). Attention Is All You Need. In *Advances in Neural Information Processing Systems*, 30.

[35] Gupta, J. K. and Brandstetter, J. (2023). Towards Multi-Spatiotemporal-Scale Generalized PDE Modeling. *Transactions on Machine Learning Research*.

[36] Hou, J., Wang, H., Xu, P., Gao, C., Liu, H., and Jing, L. (2026). Learning Neural Operators from Partial Observations via Latent Autoregressive Modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(1), 390–398.

[37] Terpin, A. and D'Andrea, R. (2025). Privileged observations enable rapid and reliable policy discovery directly in the physical world. arXiv preprint arXiv:2512.08463.

A Posterior Problem-State Formulation: Detailed Derivations

A.1 Latent world state, task family, and the canonical quotient

We spell out the minimal formal setup behind Section 3. Fix a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Let W_t be the complete latent world state at time t, let $X_t = \mathcal{O}(W_t)$ be the visible field, and let $\mathcal{T}$ be the downstream task family. For each task $\tau \in \mathcal{T}$, let Y_t^τ be its target variable taking values in a standard Borel space $\mathcal{Y}_\tau$. This assumption guarantees the existence of regular conditional laws, so for each task we may write

$$\mu_\tau(w, \cdot) := \mathbb{P}(Y_t^\tau \in \cdot \mid W_t = w)\,. \tag{16}$$

The most direct way to encode the information in W_t that matters for the entire task family is to collect these conditional laws into one object. Define

$$g^\star(w) := \Big(\mathcal{O}(w),\ \big(\mu_\tau(w, \cdot)\big)_{\tau \in \mathcal{T}}\Big), \qquad Z_t^\star := g^\star(W_t). \tag{17}$$

Equation (17) is an explicit realization of the quotient used in the main text. Indeed,

$$w \sim_{\mathcal{T}} w' \iff \mathcal{O}(w) = \mathcal{O}(w') \text{ and } \mu_\tau(w, \cdot) = \mu_\tau(w', \cdot) \quad \text{for all } \tau \in \mathcal{T}, \tag{18}$$

so $g^\star$ assigns the same label exactly to latent worlds that share the observation and are indistinguishable to the full downstream task family.

Task sufficiency is immediate from this construction. For each task τ, define the canonical kernel

$$K_\tau\big((x, (\nu_{\tau'})_{\tau' \in \mathcal{T}}), A\big) := \nu_\tau(A), \qquad A \subseteq \mathcal{Y}_\tau \text{ measurable}. \tag{19}$$

Substituting $Z_t^\star = g^\star(W_t)$ into (19) gives

$$K_\tau\big(Z_t^\star, A\big) = \mu_\tau(W_t, A) = \mathbb{P}(Y_t^\tau \in A \mid W_t)\,. \tag{20}$$

Thus every task target is conditionally determined by $Z_t^\star$; because X_t is a coordinate of $Z_t^\star$, this is the law-based form of the task-sufficiency statement in Equation (1).

The same explicit construction also proves minimality. Let $S_t = f(W_t)$ be any other task-sufficient state. By definition, for each task τ there exists a kernel $\widetilde{K}_\tau$ such that

$$\mathbb{P}(Y_t^\tau \in A \mid W_t) = \widetilde{K}_\tau\big(X_t, S_t, A\big) \qquad \text{for all } \tau \in \mathcal{T}. \tag{21}$$

Define

$$h(x, s) := \Big(x,\ \big(\widetilde{K}_\tau(x, s, \cdot)\big)_{\tau \in \mathcal{T}}\Big). \tag{22}$$

Then combining (17) and (21) yields

$$Z_t^\star = g^\star(W_t) = h(X_t, S_t) \qquad \text{a.s.} \tag{23}$$

So, within each observation fiber, every task-sufficient state S_t factors through $Z_t^\star$. Equivalently, given X_t, S_t can only refine the canonical quotient; it cannot be coarser while still preserving all downstream laws.

A.2 Posterior factorization of downstream prediction and decision

Section 3 states that single-observation simulation should factorize through the posterior over $Z_t^\star$. We now derive that statement directly from the quotient construction above. Let

$$\pi_t(B \mid x) := \mathbb{P}(Z_t^\star \in B \mid X_t = x) \tag{24}$$

be the posterior on measurable subsets B of the quotient state space. Using the kernel K_τ from (19) and the tower property, every downstream predictive law satisfies

$$\mathbb{P}(Y_t^\tau \in A \mid X_t) = \mathbb{E}\big[K_\tau\big(Z_t^\star, A\big) \mid X_t\big] = \int K_\tau(z, A)\, \pi_t(dz \mid X_t). \tag{25}$$

Equation (25) is the measure-theoretic form of the main-text factorization (4): once the observation X_t is fixed, all downstream prediction passes through the posterior on the task-sufficient latent

problem state. For a single task τ, the kernel $K_\tau(z,\cdot)$ depends on z only through its task coordinate $g_\tau(z) := K_\tau(z,\cdot)$. With $\pi_{t,\tau} := (g_\tau)_\# \pi_t(\cdot \mid x)$, Equation (25) therefore reads

$$\mathbb{P}(Y_t^\tau \in A \mid X_t = x) = \int \nu(A)\, \pi_{t,\tau}(d\nu \mid x),$$

which is the task-specific form (4): x and τ are fixed, and only the uncertainty in the task-relevant projection is integrated.

The same argument gives the decision-theoretic version used later in Section 4. Let $\ell_\tau(a,y)$ be a bounded measurable loss for task τ, where $a \in \mathcal{A}_\tau$ is a candidate decision or prediction. Define the latent-state conditional risk

$$\rho_\tau(a,z) := \int \ell_\tau(a,y)\, K_\tau(z,dy). \tag{26}$$

Then the observation-conditional risk is

$$R_\tau(a \mid x) := \mathbb{E}[\ell_\tau(a,Y_t^\tau) \mid X_t = x] = \int \rho_\tau(a,z)\, \pi_t(dz \mid x). \tag{27}$$

So the visible field affects Bayes-optimal downstream decisions only through the posterior $\pi_t(\cdot \mid x)$. Because $Z_t^\star$ contains the observation coordinate, $K_\tau(z,\cdot)$ and $\rho_\tau(a,z)$ are x-dependent kernels: within the fiber of x they equal $\mathbb{P}(Y_t^\tau \in \cdot \mid X_t = x, Z_t^\star = z)$ and $\mathbb{E}[\ell_\tau(a,Y_t^\tau) \mid X_t = x, Z_t^\star = z]$.

A.3 Countable semantic refinements and the bridge to the benchmark posterior

Let the benchmark expose a countable semantic refinement $C_t \in \mathcal{C}$ with $Z_t^\star = h(C_t)$ for a deterministic map h, and let $r_t(c \mid x) := \mathbb{P}(C_t = c \mid X_t = x)$. Then $\pi_t(\cdot \mid x) = h_\# r_t(\cdot \mid x)$, that is, $\pi_t(B \mid x) = \sum_{c \in \mathcal{C}:\, h(c) \in B} r_t(c \mid x)$, which reduces to (5) when $Z_t^\star$ is discrete, and substituting it into (27) gives

$$R_\tau(a \mid x) = \sum_{c \in \mathcal{C}} \rho_\tau\big(a, h(c)\big)\, r_t(c \mid x). \tag{28}$$

The theory can therefore speak about the minimal object $Z_t^\star$, while the benchmark supervises a countable posterior $r_t(c \mid x)$ by standard conditional distribution estimation. When $\mathcal{C}$ is not discrete, the same pushforward holds with a conditional density estimator in place of the softmax head, and the finite-class Brier barrier of Section 4.2 no longer applies verbatim.

The map h is fixed by design. Suppose $\mathcal{C}$ is finite with common support, let $K_\tau(\cdot \mid x, c) := \mathbb{P}(Y_t^\tau \in \cdot \mid X_t = x, C_t = c)$, define $c \sim_{\mathcal{T}} c'$ if $K_\tau(\cdot \mid x, c) = K_\tau(\cdot \mid x, c')$ for every $\tau \in \mathcal{T}$ and almost every x, and set $h(c) := [c]_{\mathcal{T}}$. Then $\mathbb{P}(Y_t^\tau \in \cdot \mid X_t, C_t) = K_\tau(\cdot \mid X_t, C_t)$ depends on C_t only through $h(C_t)$, so $Y_t^\tau \perp C_t \mid (X_t, h(C_t))$ for every task. Conversely, let $\tilde{h}$ be any deterministic map with $Y_t^\tau \perp C_t \mid (X_t, \tilde{h}(C_t))$ for every task. If $\tilde{h}(c) = \tilde{h}(c')$ for two labels of positive probability, both kernels equal the conditional law given $(x, \tilde{h}(c))$ for almost every x, so $c \sim_{\mathcal{T}} c'$. Hence $\tilde{h}$ merges only equivalent labels and refines h, which makes h the coarsest sufficient map up to null labels and relabeling. No function of C_t can separate full states that the catalogue has already merged, so a catalogue that merges states with different task kernels must be enriched, or the task family narrowed. Test B in Appendix D.7 instantiates this construction with a known nuisance factor.

B Why the Reusable Intermediate Object Is a Posterior: Proofs

Section 4 already contains the factorization (6), whose measure-theoretic form is derived in Appendix A.2, and the identities (7) and (8). This appendix gives the remaining steps.

Ambiguity barrier. For a fixed posterior r on a finite $\mathcal{C}$, $\|e_c - r\|_2^2 = 1 - 2r(c) + \|r\|_2^2$ is minimized by any $c \in \arg\max_{c'} r(c')$, which gives $\Gamma_{\mathrm{amb}}(r) = 1 + \|r\|_2^2 - 2\|r\|_\infty$. Since $\|r\|_2^2 \geq \|r\|_\infty^2$, $\Gamma_{\mathrm{amb}}(r) \geq (1 - \|r\|_\infty)^2$, and both sides vanish only when r is a Dirac mass. By (8), every deterministic point predictor $e_{\hat{c}(X_t)}$ therefore has excess Brier risk at least $\mathbb{E}[\Gamma_{\mathrm{amb}}(r_X)]$.

Method	Posterior exposure	Synthetic		Public			Cost
		$G_{\mathrm{rem}} \downarrow$	NLL$\downarrow$	All nRMSE$\downarrow$	GapClose$\uparrow$	Probe$\uparrow$	Rel.$\times$
`Summary`	$s_\phi(x_t) = \sum_z q(z \mid x_t) e(z)$	0.381	1.02	0.1319	0.594	0.800	1.08
`Explicit-Mix`	$\sum_z q(z \mid x_t) p_\psi(y \mid x_t, z)$	0.365	0.99	0.1307	0.611	0.807	5.60
`MC-Sample-1`	one posterior sample	0.412	1.06	0.1351	0.550	0.782	1.12
`MC-Sample-4`	four posterior samples	0.373	1.01	0.1312	0.604	0.803	4.30

Table 4: Fidelity diagnostic for the summary-conditioning implementation relative to explicit-mixture and sample-based alternatives. "Probe" denotes the public strongest hidden-metadata probe score.

Decision regret. Let $a^q = a^q_\tau(x)$ minimize $\mathcal{V}^q_\tau(\cdot, x)$ and $a^\star = a^\star_\tau(x)$ minimize $\mathcal{V}_\tau(\cdot, x)$. Then

$$\mathcal{V}_\tau(a^q, x) - \mathcal{V}_\tau(a^\star, x) = \big(\mathcal{V}_\tau - \mathcal{V}^q_\tau\big)(a^q, x) + \big(\mathcal{V}^q_\tau(a^q, x) - \mathcal{V}^q_\tau(a^\star, x)\big) + \big(\mathcal{V}^q_\tau - \mathcal{V}_\tau\big)(a^\star, x),$$

where the middle term is nonpositive by optimality of a^q and each outer term is bounded by (10), which gives (11). The bounds (12) follow from Pinsker's inequality $\|r - q\|_1 \le \sqrt{2\,\mathrm{KL}(r \,\|\, q)}$, Jensen's inequality, and (7), and from $\|v\|_1 \le \sqrt{|\mathcal{C}|}\,\|v\|_2$, Jensen's inequality, and (8).

Non-identification from a restricted task family. For $\mathcal{T}_0 \subsetneq \mathcal{T}$, let $G_{\mathcal{T}_0}(\pi) := \big(\int \rho_\tau(a, z)\,\pi(dz)\big)_{\tau \in \mathcal{T}_0,\, a \in \mathcal{A}_\tau}$. Every population objective built only from tasks in $\mathcal{T}_0$ evaluates a posterior through $G_{\mathcal{T}_0}$, so it cannot distinguish $\pi \neq \pi'$ with $G_{\mathcal{T}_0}(\pi) = G_{\mathcal{T}_0}(\pi')$, even if a task $\bar\tau \in \mathcal{T} \setminus \mathcal{T}_0$ satisfies $\int \rho_{\bar\tau}(a, z)\,\pi(dz) \neq \int \rho_{\bar\tau}(a, z)\,\pi'(dz)$ for some a. A hidden state trained only through $\mathcal{T}_0$ can therefore fit the training loss while failing as a reusable interface for $\mathcal{T}$.

C A Minimal Posterior-First Simulator: Implementation

C.1 Implementation details

Each $z \in \mathcal{Z}$ indexes one joint latent problem state of the benchmark catalogue, so one label can encode the hidden field class, operator or coefficient regime, boundary class, and memory class at once. The encoder has seven convolutional layers (1D for DS and 2D otherwise) with widths 32/64/128/256/256/256/256, each followed by GroupNorm and SiLU; the first layer preserves resolution, the others downsample by two, and global average pooling feeds a linear softmax head over the catalogue. The codebook $E \in \mathbb{R}^{|\mathcal{Z}| \times 64}$ is Xavier-initialized and learned only through the task losses, and a backbone-specific linear projection maps $s_\phi(x_t) = q_\phi(\cdot \mid x_t)^\top E$ to the feature width before it is added after the input embedding. All experiments use the log score with $\lambda_{\mathrm{post}} = 1$ and unit total task weight; Table 20 varies λ_{post}. Each configuration is trained with five seeds, and each run keeps the checkpoint with the lowest validation task-bundle risk (synthetic) or nRMSE (public). Configurations, commands, and splits are released with the code.

C.2 Summary-conditioning fidelity diagnostic

Because F_ψ is nonlinear, the summary interface (14) reproduces the posterior mixture $\sum_z q_\phi(z \mid x_t) p_\psi(y \mid x_t, z)$ only when the predictive law, or the Bayes action of the task loss, depends on x_t and q_ϕ only through $s_\phi(x_t)$. Table 4 therefore compares it with richer ways of exposing the same posterior. All variants share the posterior encoder, rollout backbone family, optimization budget, and split; `Explicit-Mix` combines conditional predictors $p_\psi(y \mid x_t, z)$ with weights $q_\phi(z \mid x_t)$, and `MC-Sample-`k averages the conditional predictions of k posterior draws.

On the synthetic benchmark, `Summary` and `Explicit-Mix` differ by only 0.016 in G_{rem} and 0.03 in future NLL, with `MC-Sample-4` in between. On the public protocol, `Explicit-Mix` reduces pooled nRMSE only from 0.1319 to 0.1307 and raises GapClose from 0.594 to 0.611, at an inference cost of 5.60 instead of 1.08 times the direct baseline. The richer interfaces are therefore slightly stronger but not qualitatively different, and the summary interface keeps most of their benefit at a fraction of their cost.

D Empirical Setup and Full Experimental Results

D.1 Metric definitions

This subsection gives the full definitions behind Section 6.1.3. For the synthetic benchmark, the average ambiguity barrier is

$$\overline{\Gamma}_{\mathrm{amb}} = \frac{1}{|\mathcal{D}_{\mathrm{test}}|} \sum_{i \in \mathcal{D}_{\mathrm{test}}} \min_{z \in \mathcal{Z}_{\mathrm{syn}}} \|e_z - r_i\|_2^2, \tag{29}$$

where $r_i(\cdot) = \mathbb{P}(z_i = \cdot \mid X_i = x_i)$ is the exact posterior for test example i, and e_z is the one-hot distribution concentrated on state z. The full task-bundle risk of method m is

$$R_{\mathrm{full}}(m) = \frac{1}{|\mathcal{D}_{\mathrm{test}}|} \sum_{i \in \mathcal{D}_{\mathrm{test}}} \sum_{\tau \in \mathcal{T}_{\mathrm{syn}}} w_\tau \, \ell_\tau\left(\hat{y}_{i,\tau}^{(m)}, y_{i,\tau}\right),$$

where $\mathcal{T}_{\mathrm{syn}}$ is the synthetic task bundle, w_τ are fixed task weights, and ℓ_τ is the task loss for target $y_{i,\tau}$. The bundle averages four tasks with equal weights $w_\tau = 1/4$: the mean-squared error of the 8-step future field normalized by the target energy, the operator cross-entropy divided by $\log 4$, the coefficient absolute error divided by the coefficient range 0.012, and the squared error of the calibrated uncertainty score. The remaining gap G_{rem} of Section 6.1.3 normalizes R_{full} between the matched direct baseline and the oracle-latent control. Future-field negative log-likelihood is

$$\mathrm{FutureNLL}(m) = -\frac{1}{|\mathcal{D}_{\mathrm{test}}|} \sum_{i \in \mathcal{D}_{\mathrm{test}}} \log p_m(y_i^{1:8} \mid x_i), \tag{30}$$

where $y_i^{1:8}$ is the 8-step future-field target and p_m is the predictive distribution of method m; deterministic predictors use Gaussian innovations calibrated on training residuals. Lower values indicate better calibrated future-field prediction. Target-wise hidden-query, calibration, and rollout-only metrics are reported in Appendix D.3.

For the public benchmark, let N_{test} be the number of test trajectories or rollout segments, T_{roll} the rollout horizon, $x_{i,t}$ the ground-truth field, and $\hat{x}_{i,t}^{(m)}$ the prediction of method m. We use

$$\mathrm{nRMSE}(m) = \frac{1}{N_{\mathrm{test}}} \sum_{i=1}^{N_{\mathrm{test}}} \sqrt{\frac{\sum_{t=1}^{T_{\mathrm{roll}}} \|\hat{x}_{i,t}^{(m)} - x_{i,t}\|_2^2}{\sum_{t=1}^{T_{\mathrm{roll}}} \|x_{i,t}\|_2^2}}, \tag{31}$$

where $\|\cdot\|_2$ is taken over all spatial locations and channels of the fields in physical units. Lower nRMSE means more accurate, and GapClose of Section 6.1.3 is computed from these values. Stratified values use the four bins of the ambiguity score in Appendix D.4. Posterior-stage calibration, coverage, and predictive NLL are reported in Appendix D.5.

The controls of Appendix D.7 use two further proper scores, for which lower is better. For a scalar parameter with posterior Q and true value f, $\mathrm{CRPS}(Q, f) = \mathbb{E}_Q|F - f| - \frac{1}{2}\mathbb{E}_Q|F - F'|$; for a predictive distribution P of fields and the true field y, the energy score is $\mathrm{ES}(P, y) = \mathbb{E}_P\|Y - y\|_2 - \frac{1}{2}\mathbb{E}_P\|Y - Y'\|_2$. Here F, F' and Y, Y' are independent draws, and both scores are estimated from posterior or rollout samples.

D.2 Synthetic exact-ambiguity benchmark: construction and splits

The synthetic benchmark is a finite exact-collision generator. Its visible fields use a Gray–Scott-style reaction–diffusion renderer [26, 27] as a standard source of smooth PDE-like patterns; exact posterior labels come from the enumerated collision construction, not approximate inverse inference. Table 5 lists the fixed data specification.

Each semantic state is a tuple of operator class, coefficient bin, and memory branch. To create exact ambiguity, the 64 states are partitioned into collision classes. States in the same class share the same rendered visible field template but induce different downstream futures and hidden-query labels. The posterior is nonuniform within each class: the dominant compatible state receives the mass shown in Table 5, and the remaining mass is distributed uniformly over the other compatible states. This construction gives the four target values of $\mathbb{E}[\Gamma_{\mathrm{amb}}(r_X)]$ used in Table 1.

Component	Synthetic exact-ambiguity setting
Visible field	$x_t \in \mathbb{R}^{1\times 64\times 64}$, scalar rendered field
Semantic catalogue	$\lvert\mathcal{Z}_{\text{syn}}\rvert = 64 = 4 \times 4 \times 4$ operator / coefficient / memory states
Main split	$40{,}000/5{,}000/5{,}000$ train/validation/test examples
Renderer	Gray–Scott-style field templates on a 64×64 grid
Ambiguity regimes	near-Dirac, mid, high, very high
Compatible states	2 / 4 / 8 / 16 per collision class
Dominant posterior mass	0.955 / 0.884 / 0.835 / 0.790; residual mass uniform in class
Expected ambiguity barrier	0.004 / 0.018 / 0.031 / 0.047
Controls	4-frame richer-context control; operator-token Dirac control
Targets	8-step future field, operator label, coefficient value, calibrated uncertainty score
Posterior label	exact $r_X(z) = \mathbb{P}(z_t = z \mid X_t)$ by finite collision-class enumeration

Table 5: Synthetic exact-ambiguity benchmark specification.

Method	Future NLL ↓	Energy score ↓	Operator Acc. ↑	Coeff. MAE ↓	Cov.@90 ↑	Full-task risk ↓
`Direct-Point`	1.48	0.412	0.41	0.173	0.54	0.462
`Direct-Dist`	1.22	0.346	0.48	0.151	0.69	0.398
`Direct-MTL`	1.24	0.349	0.66	0.108	0.71	0.342
`Direct-Large`	1.20	0.337	0.58	0.129	0.70	0.366
`Point-`z	1.18	0.331	0.71	0.094	0.74	0.333
`Posterior-`z	1.02	0.286	0.78	0.073	0.88	0.314
`RichContext-Posterior-`z	0.92	0.255	0.84	0.061	0.90	0.272
`Oracle-`z	0.81	0.226	1.00	0.000	0.92	0.223

Table 6: Full synthetic task-bundle table on the exact-collision benchmark.

Train/validation/test splits are shared across methods and use disjoint random seeds and rendered observations, while all semantic states appear in every split. The 4-frame context control adds a short history that breaks most collisions, and the operator-token control directly reveals the disambiguating operator class. These controls keep the downstream task family fixed and test whether the point-versus-posterior gap collapses when the posterior becomes effectively identifying.

D.3 Synthetic exact-ambiguity benchmark: full result tables

Tables 6–8 report the full task bundle, the ambiguity stratification, and the training-objective ablation.

Table 6 shows that the synthetic advantage of posterior-first modeling is not confined to a single headline metric. Relative to the strongest non-posterior single-field controls, `Posterior-`z improves future NLL, energy score, operator accuracy, coefficient error, and coverage simultaneously. In particular, it raises operator accuracy to 0.78, reduces coefficient MAE to 0.073, and lowers full-task risk to 0.314. The richer-context and oracle controls remain the expected upper bounds, exactly as in the main text.

Table 7 shows why the posterior advantage appears. In the near-Dirac and identifiability controls, all three single-field methods are close. As ambiguity increases, the ordering `Direct-Point` $\rightarrow$ `Point-`z $\rightarrow$ `Posterior-`z becomes steadily stronger. At very high ambiguity, `Posterior-`z reduces full-task risk from 0.401 to 0.352 relative to `Point-`z, and reduces future NLL from 1.29 to 1.12. The appendix stratification therefore agrees with the main-text mechanism result: the benefit of the posterior stage is concentrated in the regimes where the single field leaves the latent problem state most ambiguous.

Table 8 shows why rollout accuracy alone is not enough to identify the reusable state. Train-task nRMSE stays nearly unchanged across all objective families (0.028–0.030), but hidden-task score, calibration, and full-task risk improve steadily as the objective moves from direct rollout to explicit posterior recovery over the full task family. This appendix therefore reinforces the main-text conclusion: on the synthetic benchmark, posterior recovery rather than a stronger rollout stem is what accounts for the downstream improvement.

Split / ambiguity regime	`Direct-Point`		`Point-`z		`Posterior-`z	
	Full-task risk ↓	Future NLL ↓	Full-task risk ↓	Future NLL ↓	Full-task risk ↓	Future NLL ↓
Low ambiguity / near-Dirac	0.255	1.02	0.214	0.88	0.209	0.86
Mid ambiguity	0.338	1.17	0.289	1.03	0.271	0.96
High ambiguity	0.421	1.35	0.349	1.16	0.319	1.05
Very high ambiguity	0.492	1.57	0.401	1.29	0.352	1.12
4-frame context upper bound	0.214	0.86	0.187	0.78	0.184	0.77
Operator-token Dirac control	0.176	0.72	0.154	0.66	0.153	0.65

Table 7: Synthetic ambiguity stratification and identifiability controls.

Training objective family	Train nRMSE↓	Hidden↑	Long-horizon↑	Calib.↑	Full risk↓
Rollout-only direct	0.028	0.61	0.67	0.58	0.331
Rollout + auxiliary heads	0.029	0.71	0.72	0.64	0.296
Rollout + deterministic $\hat{z}$	0.029	0.77	0.76	0.68	0.271
Rollout + posterior $q(z \mid x)$, rollout task only	0.029	0.81	0.79	0.76	0.248
Full task family with posterior $q(z \mid x)$	0.030	0.84	0.82	0.83	0.231

Table 8: Synthetic rollout-only miscalibration ablation. Hidden denotes hidden-task score, Long-horizon denotes long-horizon score, Calib. denotes calibration score, and Full risk denotes full-task risk.

D.4 Public PDEBench protocol and hidden-metadata task construction

The public benchmark layer is built from PDEBench [25]. We select forward-simulation families whose released parameters, regimes, or forcing information define a hidden problem-state variable once withheld from the model input. Every public example is converted to the same single-observation protocol: the model receives one visible field x_t, while selected metadata are hidden and can only be inferred or supplied by an oracle control. Table 9 summarizes the protocol units used in our experiments.

Family	**System / field**	**Protocol / grid**	**Hidden metadata / probe**
DR	diffusion–reaction; 2D, 2 channels	1,000 seq.; 128×128; 101 frames	regime / coefficient; cls. + reg.
DS	diffusion–sorption; 1D, 1 channel	10,000 seq.; 1024 grid; 101 frames (every fifth of 501 raw)	coefficient tuple; reg.
SW	shallow water; 2D, 1 channel	1,000 seq.; 128×128; 101 frames	joint source/boundary class; cls.
INS	incompressible NS; 2D, 2 channels	1,000 seg.; 256×256 (area-averaged); 101 frames	forcing / viscosity; reg.

Table 9: Public PDEBench metadata-hidden protocol. "Seq." denotes released trajectories; "seg." denotes non-overlapping 101-frame rollout segments extracted from the released incompressible-NS trajectory stream. Direct models receive only x_t; oracle controls receive the hidden metadata. "Cls." denotes classification and "reg." denotes regression.

All public splits are trajectory-level when released trajectories are available and segment-level only for INS, where the released long simulation stream is converted into non-overlapping rollout segments before splitting. Each item has one initial field followed by a 100-step closed-loop rollout. PDEBench data are used under the CC BY 4.0 license and its code is released under the MIT license. The catalogues have 16/16/8/16 states for DR/DS/SW/INS; numeric metadata are binned into cells fitted on the training split only, and validation and test items reuse these cells unchanged. Hidden metadata are removed only from model inputs; they remain available to the evaluator and oracle control. No model sees future fields, extra trajectories, or hidden metadata beyond the protocol above.

For the ambiguity-stratified main-text figure, each evaluation sample is assigned a scalar ambiguity score $a(x_t) \in [0, 1]$, interpreted as held-out oracle-gap potential under the metadata-hidden protocol: larger values indicate that, for the same visible field, withholding metadata leaves more recoverable

benefit between the direct and oracle controls. The score is estimated on a construction subset reserved from the training partition and excluded from model training, frozen before model training, and used only for evaluation-time binning into four quantile bins. Concretely, ridge regressions fitted on one part of this subset predict future fields with and without the true metadata, a second ridge model maps initial-field descriptors to the resulting clipped relative error gap on another part, and the bin boundaries are quantiles of the score on a third part. Table 10 checks the score: in every family it correlates with the realized per-sample direct-to-oracle gap (Spearman 0.552–0.628, 0.672 pooled), and the mean gap grows from the lowest to the highest bin.

Two external probabilistic comparators are evaluated where their assumptions fit the metadata-hidden task. On SW and INS, Approximate Bayesian Neural Operators (ABNO) [28] serve as the uncertainty-aware direct surrogate: they predict rollout distributions without reconstructing the hidden problem state. On DR and DS, whose hidden metadata are low-dimensional and define a Bayesian inverse problem, we use the semi-supervised invertible neural-operator approach of Kaltenbach et al. [29]; because it returns a posterior over hidden parameters, we compose this posterior with the matched forward rollout stem of the oracle control.

Scope	Spearman $\rho(a, \Delta_{\mathrm{ora}}) \uparrow$	Mean Δ_{ora} (low) $\downarrow$	Mean Δ_{ora} (very high) $\uparrow$
DR	0.613	0.0523	0.1267
DS	0.552	0.0078	0.0216
SW	0.589	0.0189	0.0501
INS	0.628	0.0814	0.1842
All	0.672	0.0508	0.0931

Table 10: Diagnostics for the public ambiguity score used in Figure 1. Here $\Delta_{\mathrm{ora}}(x_t)$ denotes the per-sample direct-to-oracle rollout gap on the metadata-hidden protocol.

D.5 Public PDEBench metadata-hidden benchmark: complete result tables

This subsection reports the per-backbone rollout results, the strongest hidden-metadata probes, the ambiguity stratification, calibration, and the external probabilistic baselines.

Family	Backbone	Direct nRMSE	Posterior-z nRMSE	Oracle-z nRMSE	GapClose
DR	FNO	0.1760	0.1340	0.1020	0.568
DR	U-Net	0.2840	0.2150	0.1630	0.570
DR	ConvLSTM	0.2330	0.1770	0.1360	0.577
DR	Transformer	0.1980	0.1500	0.1150	0.578
DS	FNO	0.0170	0.0110	0.0070	0.600
DS	U-Net	0.0420	0.0270	0.0190	0.652
DS	ConvLSTM	0.0310	0.0200	0.0140	0.647
DS	Transformer	0.0240	0.0150	0.0100	0.643
SW	FNO	0.0510	0.0360	0.0260	0.600
SW	U-Net	0.1090	0.0740	0.0540	0.636
SW	ConvLSTM	0.0820	0.0570	0.0410	0.610
SW	Transformer	0.0670	0.0460	0.0340	0.636
INS	FNO	0.2840	0.2210	0.1770	0.589
INS	U-Net	0.4620	0.3590	0.2900	0.599
INS	ConvLSTM	0.3970	0.3080	0.2470	0.593
INS	Transformer	0.3380	0.2610	0.2080	0.592

Table 11: Per-backbone public rollout breakdown.

Table 11 shows that `Posterior-`z closes between 56.8% and 65.2% of the direct-to-oracle gap for every family and backbone, so the public improvement is not carried by a single PDE family or architecture.

Table 12 gives the strongest probe per family. Averaged separately, discrete-metadata accuracy rises from 0.704 to 0.834 and continuous-metadata R^2 from 0.553 to 0.766, covering 60% and 65% of the distance to the oracle values of 0.920 and 0.883.

Table 12: Public PDEBench hidden-metadata probe summary. Higher is better.

Family	**Probe**	**Direct probe**	**Posterior-z probe**	**Oracle-z upper bound**
DR	Acc.↑	0.726	0.845	0.932
DS	R^2↑	0.607	0.814	0.907
SW	Acc.↑	0.682	0.823	0.908
INS	R^2↑	0.498	0.718	0.858
All	Mean score ↑	0.628	0.800	0.901

Public ambiguity regime	Direct nRMSE	Posterior-z nRMSE	Oracle-z nRMSE	GapClose
Low ambiguity	0.1432	0.1237	0.0924	0.385
Mid ambiguity	0.1642	0.1316	0.0996	0.504
High ambiguity	0.1852	0.1356	0.1058	0.624
Very high ambiguity	0.2061	0.1368	0.1130	0.744

Table 13: Public ambiguity-stratified oracle-gap closure table, with bins pooled across families.

Family	Posterior-stage ECE ↓	Posterior-stage Cov.@90 ↑	Predictive Cov.@90 ↑	Predictive NLL ↓
DR	0.061	0.892	0.883	0.978
DS	0.029	0.907	0.896	0.436
SW	0.041	0.901	0.891	0.548
INS	0.072	0.889	0.879	1.117
All	0.051	0.897	0.887	0.770

Table 14: Public calibration table for the hidden posterior and its induced predictive distribution.

Table 13 shows the same monotone ambiguity effect as the main-text figure in table form: the public gap-closure ratio rises from 0.385 in the lowest-ambiguity bin to 0.744 in the highest-ambiguity bin. Table 14 shows that the gain comes with calibrated uncertainty: on the pooled public benchmark, posterior-stage ECE is 0.051, posterior-stage Cov.@90 is 0.897, and predictive Cov.@90 is 0.887.

Family	Baseline	Baseline nRMSE	Posterior-z nRMSE	Oracle-z nRMSE	Baseline GapClose	Posterior-z GapClose
SW	ABNO	0.0587	0.0532	0.0387	0.482	0.623
INS	ABNO	0.3209	0.2872	0.2305	0.353	0.594
SW+INS	ABNO	0.1898	0.1702	0.1346	0.381	0.600
DR	Invertible-NO	0.1759	0.1690	0.1290	0.500	0.573
DS	Invertible-NO	0.0200	0.0182	0.0125	0.531	0.641
DR+DS	Invertible-NO	0.0979	0.0936	0.0708	0.505	0.583

Table 15: External probabilistic baselines on the public metadata-hidden protocol: ABNO [28] on SW/INS and Invertible-NO [29] on DR/DS. Pooled rows average the nRMSE of the two families.

Table 15 compares the method with the two strongest external probabilistic baselines available under this protocol. On SW and INS, `Posterior-`z improves over ABNO in both nRMSE and gap closure, raising pooled gap closure from 0.381 to 0.600; on DR and DS, it also improves over the invertible inverse-posterior baseline, raising pooled gap closure from 0.505 to 0.583.

D.6 Additional probabilistic, supervision, and sensitivity diagnostics

The tables below specify complementary diagnostics for four practical questions: whether the posterior-first advantage survives stronger label-free uncertainty baselines, how strongly it depends on latent-label availability, how sensitive it is to the semantic taxonomy, and what computational overhead the explicit posterior stage introduces.

Two additional label-free probabilistic controls are natural complements to ABNO and Invertible-NO. `DeepEns-Direct` is an ensemble of matched direct predictors whose predictive distribution is obtained by averaging independently trained members [30]. `GateMoE-Direct` is a label-free mixture-of-experts direct predictor with a learned gating network over matched rollout experts [31]. Neither baseline receives hidden metadata or latent-label supervision. In Table 16, both improve on

`Direct-Point`, but their gap closure (0.415 and 0.478) and strongest probes (0.664 and 0.688) stay below those of `Posterior-`z (0.594 and 0.800), so generic predictive uncertainty does not recover the hidden metadata.

Method	Latent labels?	All nRMSE ↓	GapClose ↑	Strongest probe ↑	Predictive ECE ↓
`Direct-Point`	No	0.1747	0.000	0.628	0.083
`DeepEns-Direct`	No	0.1448	0.415	0.664	0.044
`GateMoE-Direct`	No	0.1403	0.478	0.688	0.052
`Posterior-`z	Yes	0.1319	0.594	0.800	0.051
`Oracle-`z	Oracle	0.1027	1.000	0.901	0.038

Table 16: Additional label-free probabilistic controls on the public metadata-hidden protocol.

To measure dependence on latent-label availability, we vary the fraction of examples for which the posterior stage receives semantic supervision while keeping the rollout supervision, backbone family, and evaluation protocol fixed. In Table 17, gap closure falls from 0.594 to 0.551, 0.429, and 0.314 as the labeled fraction drops to 50%, 25%, and 10%.

Labeled fraction	All nRMSE ↓	GapClose ↑	Strongest probe ↑	Posterior-stage ECE ↓
100%	0.1319	0.594	0.800	0.051
50%	0.1350	0.551	0.779	0.062
25%	0.1438	0.429	0.731	0.079
10%	0.1521	0.314	0.682	0.104

Table 17: Dependence of posterior-first performance on latent-label availability in the public protocol.

To delimit the countable-refinement assumption, we vary the granularity of the semantic taxonomy while keeping the visible-field protocol and rollout backbone fixed. Coarse merges nearby semantic states, default is the main-text taxonomy, and fine splits the same hidden-metadata axes more aggressively. In Table 18, the coarse taxonomy loses part of the gain (gap closure 0.526), whereas the fine taxonomy stays close to the default (0.581 vs. 0.594) but converges in four of five seeds.

Taxonomy	Mean classes / family	All nRMSE ↓	GapClose ↑	Strongest probe ↑	Successful seeds / 5 ↑
Coarse	6.5	0.1368	0.526	0.756	5/5
Default	14.0	0.1319	0.594	0.800	5/5
Fine	28.5	0.1329	0.581	0.812	4/5

Table 18: Sensitivity to semantic-taxonomy granularity and latent cardinality on the public protocol.

Finally, Table 19 records the incremental computational cost of explicit posterior recovery, which adds 7% parameters and 8% inference cost to `Direct-Point`, whereas `DeepEns-Direct` and `Explicit-Mix` cost five or more times as much at inference. Table 20 varies the posterior-stage loss weight: without the posterior term ($\lambda_{\text{post}} = 0$) gap closure drops to 0.321, and weights from 0.25 to 4 stay within 0.572–0.594.

Method	Params (M)	Rel. train cost ×	Rel. infer. cost ×	All nRMSE ↓	GapClose ↑
`Direct-Point`	31.8	1.00	1.00	0.1747	0.000
`Posterior-`z	34.1	1.13	1.08	0.1319	0.594
`DeepEns-Direct`	159.0	5.03	5.00	0.1448	0.415
`GateMoE-Direct`	48.7	1.71	1.58	0.1403	0.478
`Explicit-Mix`	78.4	3.86	5.60	0.1307	0.611

Table 19: Resource overhead of explicit posterior recovery and comparison baselines on the public protocol.

λ_{post}	All nRMSE ↓	GapClose ↑	Strongest probe ↑	Posterior-stage ECE ↓
0	0.1516	0.321	0.667	0.096
0.25	0.1335	0.572	0.786	0.057
1.0	0.1319	0.594	0.800	0.051
4.0	0.1329	0.581	0.807	0.055

Table 20: Sensitivity of the public protocol to the posterior-stage loss weight λ_{post}.

D.7 Estimator, target, interface, reuse, and stress-test controls

This subsection gives the protocols and full results of Sections 6.2.3–6.2.4. All values are means ± standard errors over five paired seeds.

Test A: posterior estimator and summary. We follow PDEArena's conditioned 2D Navier–Stokes task with the U-Net$_{\text{mod}}$-64 simulator and its scalar-conditioning path [35], hide the buoyancy parameter f, and report in-distribution clean results; the PDEArena code and its conditioned Navier–Stokes data are released under the MIT license. All rows share the target, data, simulation budget, a 128-dimensional summary, the frozen simulator, and a 64-sample posterior-mixture rollout in which each draw of f is fixed throughout its rollout. Rows 2–3 use one common frozen CNN summary, the PCA summary is fitted on the training split with matched dimension, and row 4 is our jointly learned implementation. Table 21 reports the CRPS of the posterior over f, the coverage of its central 90% interval, and the joint Energy score over the final quarter of the rollout. Replacing PCA by the frozen CNN changes CRPS/Energy from 0.2116/0.1517 to 0.1916/0.1459; with that CNN fixed, conditional-diffusion NPE [21] gives 0.1898/0.1464, slightly better in CRPS and slightly worse in Energy than the rational-quadratic spline NPE [19].

Posterior estimator	Observation summary	f-CRPS ↓	Cov.@90	Rollout Energy ↓
RQS-NPE	train-split PCA	0.2116 ± 0.0022	$86.8 \pm 0.7\%$	0.1517 ± 0.0009
RQS-NPE	common frozen CNN	0.1916 ± 0.0017	$89.6 \pm 0.4\%$	0.1459 ± 0.0008
Conditional-diffusion NPE	same frozen CNN	0.1898 ± 0.0018	$90.8 \pm 0.4\%$	0.1464 ± 0.0008
RQS-NPE	jointly learned CNN	0.1874 ± 0.0014	$90.4 \pm 0.5\%$	0.1438 ± 0.0014

Table 21: Test A: modern SBI estimators with fixed and learned summaries on PDEArena Navier–Stokes with hidden buoyancy.

Test B: inference target and deployment. We reuse the synthetic exact-collision benchmark with $|\mathcal{Z}^\star| = 64$ and add a balanced observation nuisance $U \in \{1, \dots, 8\}$, giving $|\mathcal{C}| = 512$; U enters no target or transition kernel, so the construction of Appendix A.3 gives $h(z, u) = z$. With the same learned-summary NPE family and nested training pools, we compare inference on the full C followed by the exact pushforward $h_{\#}q(z \mid x) = \sum_u q(z, u \mid x)$, an operator-only slice completed by the conditional training prior, and direct $Z^\star$ inference. The $Z^\star$-score is the squared L_2 distance between the inferred and exact posteriors over $Z^\star$; Sims@0.025 is the training budget at which this score reaches 0.025 on nested pools of up to 160k simulations; and the risk is the all-task risk R_{full} of Appendix D.1 under MAP or mixture deployment of the same frozen posterior and task kernel, with MAP taken after pushforward. Table 22 reports the results at 40k training simulations. Full-C pushforward and direct $Z^\star$ inference are asymptotically equivalent under correct specification and consistency, so their difference isolates a finite-budget target-alignment effect.

NPE target	$Z^\star$-score ↓	All-task risk, MAP ↓	All-task risk, Mix ↓	Sims@0.025 ↓
Full C, then $h_{\#}q$	0.0294 ± 0.0007	0.3507 ± 0.0014	0.3331 ± 0.0016	79.4 ± 1.8k
Operator-only slice	0.0765 ± 0.0017	0.3825 ± 0.0020	0.3668 ± 0.0023	>160k (0/5 reached)
Direct $Z^\star$	0.0187 ± 0.0005	0.3391 ± 0.0014	0.3196 ± 0.0015	30.4 ± 1.1k

Table 22: Test B: target selection and posterior deployment with a fixed learned-summary NPE family.

Matched latent-diffusion controls. We adapt the autoencoder–latent-diffusion recipe of Rozet et al. [24] to the metadata-hidden PDEBench splits and match the autoencoder, denoiser size, training and sampling steps, and compute. `Point-LDM`, `Posterior-First LDM`, and `Oracle-LDM` share one state-conditional checkpoint; `Direct-LDM` and `Direct-LDM+MTL` use null hidden-state conditioning; and `Oracle-LDM` receives the true state, paralleling the supplied simulation parameters of [24]. `Direct-LDM+MTL` trains the observation trunk of the direct denoiser with the same labels, label fraction, q-head architecture, and scoring loss, but withholds q from rollout. `Point-LDM` passes the maximum-posterior state of the shared frozen q to every rollout, whereas `Posterior-First LDM` draws $c \sim q(c \mid x)$ for each rollout. We use 16 rollout samples per input, the nRMSE of Appendix D.1, a per-field-normalized Energy score, the ambiguity strata of Appendix D.4, and the Shallow-Water boundary Brier score $(p_{\text{boundary}} - \mathbf{1}_{\text{boundary}})^2$, read from the trained q without a query-specific head or loss. In Table 23, `Posterior-First LDM` closes 76.0% of the highest-ambiguity Direct-to-Oracle Energy gap and 59.0% of the pooled nRMSE gap. `Direct-LDM+MTL` closes about 4% of the pooled nRMSE gap, and its Energy improvement is 9.3% of the Direct-to-Posterior improvement; under identical label access the boundary Brier falls from 0.1317 to 0.1169 (paired difference 0.0148 ± 0.0021). With the shared q and checkpoint, the Point-to-Posterior Energy gap grows from 0.0052 in the lowest-ambiguity stratum to 0.0435 in the highest.

Method	Hidden state in rollout	nRMSE ↓	Energy ↓	High-amb. Energy ↓	Boundary Brier ↓
`Direct-LDM`	none	0.1744 ± 0.0014	0.1604 ± 0.0018	0.2420 ± 0.0031	–
`Direct-LDM+MTL`	q is auxiliary only	0.1716 ± 0.0013	0.1569 ± 0.0017	0.2354 ± 0.0030	0.1317 ± 0.0025
`Point-LDM`	one state from shared q	0.1508 ± 0.0012	0.1398 ± 0.0014	0.1920 ± 0.0026	0.1847 ± 0.0041
`Posterior-First LDM`	mixture from shared q	0.1323 ± 0.0010	0.1226 ± 0.0012	0.1485 ± 0.0021	0.1169 ± 0.0019
`Oracle-LDM`	true state	0.1030 ± 0.0008	0.0978 ± 0.0009	0.1190 ± 0.0016	0.0000 ± 0.0000

Table 23: Matched latent-diffusion controls on the metadata-hidden PDEBench protocol.

Leave-one-task-out transfer. We use four-fold leave-one-task-out transfer over the four synthetic targets: the 8-step future field, the operator label, the coefficient value, and the calibrated uncertainty score. For a held-out task $\tau^\star$, the Posterior-only encoder is trained only with the proper score, and the Joint-minus-held-out encoder also uses the other three task losses; both are then frozen, so the loss of $\tau^\star$ never reaches either encoder. The same 10% of the labels of $\tau^\star$ train a conditional head $p_{\psi,\tau^\star}(y \mid x, c)$ shared by the posterior variants. Mixture marginalizes this head over q, MAP uses the mode of the Posterior-only q, Oracle uses the true c, and Direct learns $x \mapsto y$ without C supervision. The risk $R_{\text{full}}^{\text{LOTO@10\%}}$ uses the losses and weights of Appendix D.1, macro-averaged over the four folds and five paired seeds. In Table 24, Posterior-only beats MAP on all four held-out targets; on the future-field fold, NLL and Energy fall from 1.310 and 0.354 to 1.147 and 0.317. The paired difference between the Posterior-only and Joint-minus-held-out risks is 0.0060 ± 0.0026 (t_4 95% interval $[-0.0013, 0.0133]$, from unrounded seed-wise differences), and Posterior-only retains 96.4% of the Direct-to-Oracle gap closure of Joint-minus-held-out. The task-specific head is retrained, and zero-shot whole-model or out-of-ontology reuse is not tested.

Held-out interface	q-ECE ↓	$R_{\text{full}}^{\text{LOTO@10\%}}$ ↓
Direct	–	0.5210 ± 0.0048
MAP, Posterior-only q_ϕ	same q	0.3954 ± 0.0053
Mixture, Posterior-only q_ϕ	0.0392 ± 0.0020	0.3618 ± 0.0032
Mixture, Joint-minus-held-out q_ϕ	0.0408 ± 0.0018	0.3558 ± 0.0055
Oracle c	–	0.2788 ± 0.0024

Table 24: Leave-one-task-out reuse of a frozen posterior encoder with 10% of the held-out task's labels.

Compound stress test. We follow PDEArena's conditioned Navier–Stokes task, U-Net$_{\text{mod}}$-64, scalar-conditioning path, and published buoyancy ranges $f \in [0.2, 0.5]$ in distribution and $[0.1, 0.2) \cup (0.5, 0.6]$ out of distribution [35], together with the 50%/75% training/test patch missingness of POBench-PDE with patch size 4 [36]. We hide f, add Gaussian noise with standard deviation up to $0.05\sigma_c$ to the initial field, where σ_c is the channel scale, replace the discrete head with a four-layer conditional spline flow trained by negative log-likelihood on in-distribution data and then frozen, and evaluate unseen 200-step rollouts. All variants use the same mask-aware input and a matched 1:1

mixture of clean and corrupted training inputs. Within each seed, the posterior-mean and MC-64 variants share the same frozen $q_\phi(f \mid x)$ and the same simulator trained with the true f, and differ only in conditioning on $\mathbb{E}_q[f]$ or on 64 posterior draws, each fixed throughout its rollout; Oracle supplies the true f, and Direct omits f. Neither out-of-distribution parameters nor long trajectories enter training. Table 25 reports the joint Energy score over the final quarter (steps 151–200) on training-standardized fields; MC-64 uses the unbiased ensemble estimator, and deterministic rows use the exact Energy of a degenerate distribution. The seed-wise paired improvements of MC-64 over the posterior mean are 0.0055 ± 0.0013, 0.0294 ± 0.0024, and 0.0418 ± 0.0037 across the three regimes, while the flow's central 90% coverage decreases from $90.4 \pm 0.5\%$ to $83.6 \pm 1.4\%$. Comparing ID clean with ID sparse+noise measures the joint change associated with sparse and noisy sensing, and comparing ID sparse+noise with OOD sparse+noise measures the additional change associated with parameter shift under the same sensing; the test does not separate the individual stressors.

Rollout interface	ID clean	ID sparse+noise	OOD sparse+noise
Direct	0.1762 ± 0.0018	0.2317 ± 0.0030	0.2789 ± 0.0040
Posterior mean $\mathbb{E}_q[f]$	0.1493 ± 0.0016	0.2036 ± 0.0027	0.2476 ± 0.0036
Posterior-first (MC-64)	0.1438 ± 0.0014	0.1742 ± 0.0022	0.2058 ± 0.0029
Oracle f	0.1295 ± 0.0012	0.1511 ± 0.0019	0.1774 ± 0.0025

Table 25: Compound stress test with a continuous hidden parameter, sparse noisy observations, parameter shift, and unseen 200-step rollouts (final-quarter Energy, lower is better).

E Use of Large Language Models

LLMs were used only to assist with writing and presentation: checking technical terminology, correcting grammar, translating from the authors' native language, and improving sentence structure while preserving the scientific content. They were not used for research ideation, experimental design, data analysis, or any content that constitutes an intellectual contribution. All scientific insights, methodological decisions, and conclusions are the authors' own.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and Introduction state the posterior-first claim, scope it to the task-to-target-to-interface paradigm and its theory rather than a new estimator or backbone, and summarize the evidence; Sections 3–6 provide the formulation, theory, method, and experiments, and Section 7 bounds the claim.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 7 discusses the single-observation scope, the assumed map h without automatic quotient discovery, supervised hidden-state labels, the summary approximation to explicit mixtures, the scope of the stress test and of encoder reuse, task-family and stability assumptions, and the absence of closed-loop validation.

   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: Sections 3–4 state the objects, assumptions, and identities, including the construction of h. Appendices A–B give the quotient construction, posterior factorization, proper-scoring identities, ambiguity barrier, and utility bounds with proofs.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Sections 5–6 and Appendices C–D specify the architecture, objective, data protocols, catalogues, metrics, model selection, and the protocols of the added controls; the released code provides the configurations and commands.

   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: The code is at https://github.com/VinceWangVW/Posterior-First with installation instructions, configurations, and commands; PDEBench and PDEArena data are public (Appendix D).

   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: Sections 5–6 and Appendices C–D give the splits, catalogues, objectives, loss weights, model selection, and metrics; exact hyperparameters, configurations, and commands are released with the code.

   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: The added controls (Section 6.2.3, Appendix D.7) report means $\pm$ standard errors over five paired seeds and seed-wise paired differences; the synthetic and public summary tables report five-seed means.

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: [No]

   Justification: Appendix D.6 reports parameter counts and relative training and inference costs, but the paper does not state the hardware type, memory, or wall-clock time of each experiment.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work uses synthetic and public PDE simulation data and involves no human subjects or sensitive data.

   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: Section 7 discusses more reliable scientific surrogate modeling and decision-making, such as feedback-based flow control; a potential negative impact is over-reliance on inferred hidden physics outside validated task families, which the stated limitations and scoped claims address.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: We release no model or dataset with a high risk of misuse.

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: PDEBench (data CC BY 4.0, code MIT) and PDEArena (code and Navier–Stokes data MIT) are cited, with licenses in Appendix D.

Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: The released code is documented in its repository and distributed under the MIT license.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The work involves no crowdsourcing or human subjects.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The work involves no 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: LLMs were used only for writing and editing (Appendix E), not as part of the method.

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.