# BAYESIAN FILTERING IN PHYSICAL SYSTEMS VIA TEST-TIME TRAINED FLOW MATCHING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Bayesian filtering provides a principled framework for online state estimation under uncertainty. Applying it to high-dimensional systems with complex posterior distributions remains challenging. Recent generative models, such as flow matching, have shown potential in Bayesian filtering. However, existing approaches either propagate a finite sample ensemble or repeatedly sample a state trajectory, coupling online cost to the ensemble size or trajectory sampling budget. To address this, we propose a new perspective of *directly encoding the evolving distribution into flow matching model weights*, namely, the Belief Flow Filter (BFF). It is a generative filtering framework that updates model weights via gradient descent at test time to track the posterior evolution. BFF carries the belief forward in generative model weights and draws samples from the updated model without restricting the posterior to a Gaussian family. A Bayes-derived loss motivates the update design, where an idealized expressivity result and a conditional, population-level bound connect the outer training loss to filtering error. The deployed finite-capacity update amortizes a specified observation model and has no guarantee of exact Bayesian inference under an unseen likelihood. Across five physical systems, BFF achieves the lowest reconstruction error and continuous ranked probability score (CRPS) on Kuramoto–Sivashinsky (KS) and Burgers' and ranks among the top two methods on all four metrics for Navier–Stokes (NS) flow with uniform sensors. BFF also achieves the lowest reconstruction error, spectrum error, and miscalibration area (MA) for NS flow with a single moving sensor, and the lowest gradient error, spectrum error, and MA for synthetic tokamak profile estimation from diagnostic measurements.

## 1 INTRODUCTION

Sequential Bayesian filtering infers the posterior distribution of a system state, or belief state, from noisy observations. It is central to physical applications ranging from weather forecasting to magnetically confined nuclear fusion. Traditional Bayesian filters (Kalman, 1960; Evensen, 2009b; Gordon et al., 1993) rely on Gaussian assumptions or discrete particle ensembles, which struggle in the presence of complicated dynamics and posteriors. Crucially, the Gaussian assumption fails to capture the multi-modality or long tail in the posterior, while particle-based methods tend to suffer from the curse of dimensionality and can require a prohibitively large number of particles to represent the full posterior in high-dimensional state spaces.

Recent flow-based generative models (Lipman et al., 2023; Ho et al., 2020) provide a scalable free-form probabilistic modeling framework. Yet, their application to Bayesian filtering problems remains limited. Rozet & Louppe (2023) frames filtering as posterior sampling over a state trajectory. Repeated trajectory-level sampling can be computationally expensive in online filtering. Alternatively, Chen et al. (2025) and Bao et al. (2024) use recursive sampling-based updates. Their finite ensembles approximate the filtering distribution, with accuracy and cost depending on the ensemble and guidance choices. Concurrent sequential flow matching (Huang et al., 2026) instead uses samples from the previous posterior to warm-start the next probability flow. These are complementary recursive approaches. Our contribution is a persistent generator whose updated weights encode the belief.

Can we maintain the recursive nature of Bayesian filtering while alleviating the bottleneck of representing the posterior? Here, we introduce the Belief Flow Filter (BFF), whose key insight is *using flow matching model weights to encode the posterior distribution*. A learnable surrogate



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

loss updates the flow matching weights through test-time training (TTT) (Sun et al., 2025; 2020), tracking posterior evolution at each physical step. Furthermore, we train the surrogate through an unrolled outer objective that penalizes the beliefs produced by recursive deployment. Under stated regularity assumptions, its population loss bounds an average output-level filtering error over the training distribution and horizon.

We evaluate BFF on sparsely observed partial differential equations (PDEs), long-horizon Lorenz-63 filtering, and synthetic tokamak trajectories based on discharge 152444 of the Experimental Advanced Superconducting Tokamak (EAST). Compared with Kalman, ensemble-score, deterministic, and generative baselines, BFF achieves the lowest reconstruction error and CRPS on one-dimensional (1D) KS and Burgers' systems and ranks among the top two methods in all four metrics on two-dimensional (2D) NS flow with uniform sensors. Score-based data assimilation (SDA) has lower 1D MA and lower NS reconstruction error and CRPS, while the ensemble Kalman filter (EnKF) has lower Burgers' and NS spectrum errors. With a single moving sensor, BFF has the lowest reconstruction error, spectrum deviation, and MA. On synthetic tokamak profiles, it has the lowest gradient error, spectrum error, and MA, while the Transformer has the lowest reconstruction error.

Our main contributions are three-fold.

(1) A recursive filtering architecture that stores the evolving belief in *flow-generator weights* and updates those weights as observations arrive.

(2) A Bayes-derived motivation for the surrogate update, an expressivity result for an idealized residual-gradient interface, and a conditional population bound linking outer training loss to distribution-level error. These results do not guarantee exactness of the finite, probe-restricted implementation.

(3) Experiments showing competitive reconstruction, spectral accuracy, and uncertainty calibration across physical systems with chaotic dynamics, sparse observations, and nonlinear observation models. Ablations and stress tests identify sensitivity to the update architecture, sampling resolution, training seed, and observation protocol (Appendix K). The code is open-sourced here.

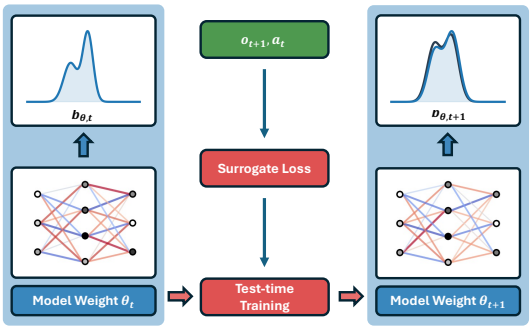



Figure 1: **An overview of our proposed BFF.** Blue frames denote the flow matching model parameterized by $\theta$ at different physical times. The model weights $\theta_t$ parameterize the posterior distribution $b_{\theta_t}$, and evolve ($\theta_t \to \theta_{t+1}$) given observation $o_{t+1}$ and action $a_t$, creating the next posterior $b_{\theta_{t+1}}$. The update is executed by a gradient step on a learnable surrogate loss $\ell_\phi$ that takes $\theta_t$, $o_{t+1}$, and $a_t$ as inputs.

## 2 Background

### 2.1 Problem Setup

Consider a discrete-time system with physical state $s_t \in \mathcal{X} \subseteq \mathbb{R}^n$, where $\mathcal{X}$ is the state space. Examples include fluid velocity and pressure fields or tokamak temperature and density profiles. Subject to a control action $a_t \in \mathcal{A} \subseteq \mathbb{R}^m$, the state evolves to $s_{t+1}$ governed by the environmental transition dynamics $p(s_{t+1} \mid s_t, a_t)$. However, in a partially observable setting, the exact physical state remains hidden. Instead, we only have access to noisy and potentially sparse observations $o_t \in \mathcal{O} \subseteq \mathbb{R}^k$ (typically $k \ll n$), generated by an observation model $p(o_t \mid s_t)$. Here, $\mathcal{A}$ and $\mathcal{O}$ denote the action and observation spaces, respectively. We adopt the standard Markov assumption that the transition depends only on the current state and action, and the observation depends only on the current state. These conditional-independence assumptions give the recursive filtering updates below.

Given the observation-action history $h_t := (o_{1:t}, a_{1:t-1})$, the belief state is the posterior distribution $b_t(s) := p(s_t = s \mid h_t)$. The filtering problem can thus be viewed as recursively updating this belief state $b_t$ into $b_{t+1}$ upon executing action $a_t$[1]and receiving a new observation $o_{t+1}$. This gives the

following prediction and update steps.

$$\bar{b}_{t+1}(s) = \int p(s \mid s', a_t) b_t(s') \, \mathrm{d}s' \qquad \text{(Prediction Step)} \tag{1}$$

$$b_{t+1}(s) = \frac{\bar{b}_{t+1}(s) p(o_{t+1} \mid s)}{\int \bar{b}_{t+1}(s) p(o_{t+1} \mid s) \, \mathrm{d}s} \qquad \text{(Update Step)} \tag{2}$$

where $\bar{b}_{t+1}$ is the propagated *prior*, the distribution of where the system would land at $t+1$ before the new observation arrives, whereas $b_{t+1}$ is the Bayesian *posterior* obtained by updating the propagated prior with the latest observation. Equations (1) and (2) together define a recursive filtering operator $K : \mathcal{P}(\mathcal{X}) \times \mathcal{O} \times \mathcal{A} \mapsto \mathcal{P}(\mathcal{X})$ that maps the current belief state, together with the latest observation and action, to the next belief state. Here, $\mathcal{P}(\mathcal{X})$ is the space of probability measures on $\mathcal{X}$. In the following, we abbreviate $K(b_t, o_{t+1}, a_t)$ as $Kb_t$ whenever the new information $(o_{t+1}, a_t)$ is clear from context.

$$b_{t+1}(s) = Kb_t(s) \coloneqq \frac{1}{Z_{t+1}} p(o_{t+1} \mid s) \int p(s \mid s', a_t) b_t(s') \, \mathrm{d}s', \tag{3}$$

where $Z_{t+1} = \int p(o_{t+1} \mid s) \int p(s \mid s', a_t) b_t(s') \, \mathrm{d}s' \, \mathrm{d}s$ is the normalization constant ensuring $b_{t+1}$ integrates to one. We represent the belief with the weights of a flow model and learn a test-time weight update that approximates the recursive operator $K$.

### 2.2 Neural Representation of Belief States

Flow-based generative models (Ho et al., 2020; Lipman et al., 2023) represent belief distributions through a learned velocity field $u_\theta(s^{(\tau)}, \tau)$ that defines a probability flow ordinary differential equation (ODE) $\mathrm{d}s^{(\tau)}/\mathrm{d}\tau = u_\theta(s^{(\tau)}, \tau)$ for $\tau \in [0, 1]$. Integrating this ODE induces a flow map $\phi^{u_\theta} : \mathcal{X} \to \mathcal{X}$ that transports a Gaussian base distribution[2] $p^{(0)}$ to the model distribution $p^{(1)}$. Throughout, superscripts in parentheses (*for example*, $s^{(\tau)}, p^{(\tau)}$) index *flow time* internal to the generation process, while subscripts (*for example*, $\theta_t, b_t$) index *physical time* of the filtering process.

Fitting $\theta$ to a target distribution $q$ uses the conditional flow matching (CFM) loss (Lipman et al., 2023; 2024)

$$\mathcal{L}_{\text{CFM}}(\theta; q) \;=\; \mathbb{E}_{\tau \sim U(0,1),\, s^{(0)} \sim p^{(0)},\, s^{(1)} \sim q} \left[ \left\| u_\theta(s^{(\tau)}, \tau) - (s^{(1)} - s^{(0)}) \right\|^2 \right], \tag{4}$$

with $s^{(\tau)} = \tau s^{(1)} + (1 - \tau) s^{(0)}$, uniform flow time $\tau \sim U(0, 1)$, expectation $\mathbb{E}$, and Euclidean norm $\|\cdot\|$. We write $\mu_q$ for the resulting joint distribution of $(s^{(\tau)}, \tau)$, the flow matching probability path of $q$. If the model can represent the target marginal velocity and the flow ODE generates the corresponding probability path, attaining the population minimum yields $p^{(1)} = q$.

At physical time $t$, weights $\theta_t$ in parameter space $\Theta$ parameterize $u_{\theta_t}$ and define the mapping

$$\Phi : \Theta \to \mathcal{P}(\mathcal{X}), \quad \theta_t \mapsto b_{\theta_t}, \tag{5}$$

where $b_{\theta_t}$ is our model's approximation of the true belief $b_t$. The weights $\theta_t$ specify the belief through the flow map. To draw a state sample from $b_{\theta_t}$, we draw Gaussian noise and integrate the ODE using $u_{\theta_t}$.

### 2.3 Bayesian Filtering Update in Neural Network Weight Space

At filtering step $t$, the flow matching weights $\theta_t$ induce a belief $b_{\theta_t}$. As recursive Bayesian filtering updates beliefs in distribution space, our filtering algorithm needs to update neural network weights

$$b_{t+1} = K(b_t, o_{t+1}, a_t) \Leftrightarrow \theta_{t+1} = \mathcal{K}(\theta_t, o_{t+1}, a_t), \tag{6}$$

where $\mathcal{K}$ denotes an ideal weight-space lift of $K$, assuming that the current and updated beliefs are representable. This correspondence is not a guarantee for the finite deployed update. Given $\theta$, we can

[1] Filters must account for action when control is exerted on the physical system. For example, plasma in tokamaks is controlled, and our experiments also include this case. When action is not present, the argument $a$ is simply omitted.

[2] Flow matching can also use other base distributions. We use a Gaussian base throughout.

sample from $b_\theta$ by integrating the flow ODE from Gaussian noise. Finding weights whose induced distribution matches a target $Kb_{\theta_t}$ generally requires optimization.

In this subsection, we derive an ideal velocity-matching loss and an empirical estimator that motivate the learned update. Exact realization requires a sufficiently expressive model and attainment of the ideal minimum. The finite-sample estimator and the deployed learned surrogate introduce additional approximation gaps.

**Bayesian filtering step as an ideal loss minimizer.** We reformulate the belief-space target $Kb_{\theta_t}$ as a velocity-matching objective in weight space, define $\mathcal{K}$ as its minimizer, and use its target structure to motivate a learned test-time update. We fit the updated-posterior velocity at points sampled along the propagated-prior flow path. This gives the *ideal filtering loss*

$$\ell^\star(\theta) \;=\; \mathbb{E}_{(s,\tau)\sim\mu_{\bar{b}_{\theta_{t+1}}}}\left[\left\|u_\theta(s,\tau) - u_{Kb_{\theta_t}}(s,\tau)\right\|^2\right], \tag{7}$$

where $u_{Kb_{\theta_t}}$ is the updated-posterior marginal velocity and $\bar{b}_{\theta_{t+1}}$ is the propagated model prior from Equation (1). The path law $\mu_{\bar{b}_{\theta_{t+1}}}$ follows the construction in Section 2.2. The current belief, observation, action, and hence the sampling law and target are fixed when optimizing $\theta$. The target is $Kb_{\theta_t}$ because the update starts from the model's current belief $b_{\theta_t}$. The outer training of $\phi$ targets the discrepancy between the recursively produced beliefs and the data-conditioned beliefs. Theorem 3.1 bounds the resulting population operator error under its assumptions, but does not guarantee that training eliminates this discrepancy. When the minimum is attained, we define the ideal weight-space update $\mathcal{K}$ by choosing a minimizer,

$$\mathcal{K}(\theta_t, o_{t+1}, a_t) \;\in\; \arg\min_\theta\; \ell^\star(\theta), \tag{8}$$

When the posterior velocity is realizable and the flow ODEs are well posed and generate their specified marginals, attaining zero loss yields the updated posterior. Gaussian smoothing ensures that the prior path covers the posterior path for $0 < \tau < 1$, as detailed in Appendix F.1. Otherwise, an attained minimizer gives a best fit under the prior-path weighting within the model class. If the infimum is not attained, only approximate minimization is available.

**Empirical filtering loss with propagated-prior anchors.** However, we cannot evaluate $\ell^\star$ directly because the target velocity $u_{Kb_{\theta_t}}$ is unknown. The Bayesian filtering and flow matching structures (elaborated in Appendix F.1) yield an *empirical filtering loss* $\hat{\ell}$ whose targets combine propagation and a likelihood correction. Proposition 1 relates its expectation to $\ell^\star$. For fixed realized anchors, it consists of a model branch $F_N$ varying with the candidate weights $\theta$ and a target branch $G_N$ fixed by $(\theta_t, o_{t+1}, a_t)$ and the sampled targets. Their anchor dependence is suppressed in the notation. Both branches are evaluable without the unknown $u_{Kb_{\theta_t}}$ and take the form

$$\hat{\ell}(\theta) \;:=\; \|F_N(\theta) - G_N(\theta_t, o_{t+1}, a_t)\|^2, \tag{9}$$

$$F_N(\theta) \;=\; \frac{1}{\sqrt{N}}\big(u_\theta(s_i^{(\tau_i)}, \tau_i)\big)_{i=1}^N,$$

$$G_N(\theta_t, o_{t+1}, a_t) \;=\; \frac{1}{\sqrt{N}}\Big(\underbrace{s_i^p - s_i^{(0)}}_{\text{propagation}} + \frac{1-\tau_i}{\tau_i}\,\underbrace{\nabla_s \log \bar{p}^{(\tau_i)}(o_{t+1}\mid s)\big|_{s=s_i^{(\tau_i)}}}_{\text{likelihood}}\Big)_{i=1}^N, \tag{10}$$

where $()_{i=1}^N$ denotes the concatenation of $N$ vectors. The constituents of $F_N, G_N \in \mathbb{R}^{N\times n}$ are defined as follows. For each anchor $i \in \{1,\dots,N\}$, $(s_i^{(0)}, \tau_i)$ are the standard flow-matching anchor noise and time, and $s_i^p \sim \bar{b}_{\theta_{t+1}}$ is sampled from the propagated model prior of Equation (1). Together, they form the path point $s_i^{(\tau_i)} = \tau_i s_i^p + (1-\tau_i)s_i^{(0)}$, thus defining the *propagation* term and the *likelihood* term in Equation (10). In the likelihood term, path-smoothed likelihood $\bar{p}^{(\tau)}(o_{t+1}\mid s) \propto \mathbb{E}_{s'\sim\bar{b}_{\theta_{t+1}}}[\,p(o_{t+1}\mid s')\,\mathcal{N}(s;\tau s', (1-\tau)^2 I)\,]$ (normalized by integrating out $o_{t+1}$) is evaluated by self-normalized Monte-Carlo with $N_j$ propagation samples. Here, $\mathcal{N}(s;m,\Sigma)$ denotes a Gaussian density with mean $m$ and covariance $\Sigma$, and $I$ is the identity matrix. The details are in Appendix F.1.

Each propagation term is obtained by sampling from $b_{\theta_t}$, applying the transition model, and subtracting the Gaussian anchor noise. Conditional on the path point and flow time, its mean is the

propagated-prior velocity. With the exact path-smoothed likelihood, adding the likelihood term gives a conditionally unbiased target for the updated-posterior velocity. Monte-Carlo evaluation of the likelihood introduces the finite-$N_j$ bias bounded below.

**Proposition 1** (Empirical filtering loss, informal)**.** *With independent anchors and independent extra propagation samples per anchor, assume finite fourth moments of the exact residual and the conditional first-, second-, and fourth-moment bounds for the likelihood-target error stated in Appendix F.1. Then $\hat{\ell}$ estimates $\ell^\star$ up to a $\theta$-independent constant $C$, with bias and variance*

$$\left|\mathbb{E}[\hat{\ell}(\theta)] - \ell^\star(\theta) - C\right| \;\le\; \frac{L_1}{N_j}, \qquad \mathrm{Var}\big(\hat{\ell}(\theta)\big) \;\le\; \frac{1}{N}\Big(\sigma^2 + \frac{c_5}{N_j}\Big),$$

*where* Var *denotes variance, and constants $L_1, c_5, \sigma^2$ are determined by these moment bounds and the model class. These assumptions are not implied by finite prior variance alone. In particular, the Monte-Carlo bounds need not hold uniformly as the flow time approaches an endpoint.*

The value and gradient approximation errors of this analytical estimator are distinct. The finite-$N_j$ bias need not vanish by increasing the anchor count $N$ alone. This estimator motivates the learned surrogate but is not evaluated during deployed BFF updates. The full assumptions and bounds are deferred to Appendix F.1.

## 3 Test-time Training of Flow Matching Filters

BFF approximates recursive filtering by updating flow matching weights $\theta$ through TTT using a surrogate loss $\ell_\phi$ with learnable parameters $\phi$. Training involves two loops. The inner loop updates $\theta$ via a $\phi$-dependent surrogate loss (Section 3.1 and Algorithm 2). The outer loop updates $\phi$ via the outer loss (Section 3.2 and Algorithm 1). The implementation details are in Appendix D.

### 3.1 Surrogate Loss for Test-time Training

The empirical filtering loss $\|F_N(\theta) - G_N(\theta_t, o_{t+1}, a_t)\|^2$ estimates the ideal filtering loss $\ell^\star$ under the stated assumptions, but evaluating $G_N$ at each step requires querying the dynamics and observation models. We instead learn an update for a specified model family, shifting this computation to offline training to support applications with strict online latency budgets, such as tokamak diagnosis. This motivates a *learnable* surrogate loss

$$\ell_\phi(\theta; \theta_t, o_{t+1}, a_t) \;=\; \|f_\phi(\theta) - g_\phi(\mathrm{sg}[\theta_t], o_{t+1}, a_t)\|^2_{\mathcal{H}}\,, \tag{11}$$

where $\mathcal{H}$ is the shared latent space, sg is the stop-gradient operator, and $f_\phi$ and $g_\phi$ are two *learnable* projection heads[3] which are motivated by $F_N$ and $G_N$, respectively. Proposition 3 gives an approximation bound if the heads approximate their specified targets. The outer training objective does not impose or verify those head-level approximation conditions. Thus the trained $\ell_\phi$ is not proven to match $\hat{\ell}$ or the ideal Bayes-derived loss. In practice, we use a single test-time gradient step on $\ell_\phi$ to update $\theta_{t+1}$,

$$\theta_{t+1} \;=\; \mathcal{K}_\phi(\theta_t, o_{t+1}, a_t) \;=\; \theta_t - \eta\, \nabla_\theta\, \ell_\phi(\theta; \theta_t, o_{t+1}, a_t)\big|_{\theta=\theta_t}, \tag{12}$$

where $\eta$ is the inner-loop learning rate and $\mathcal{K}_\phi$ is the learned weight update approximating $\mathcal{K}$. As Figure 2 shows, our implementation uses the diffusion transformer (DiT) backbone (Peebles & Xie, 2023), augmented with a per-layer TTT module inserted after each self-attention block. The TTT module's weights are updated as $\theta$, while the rest of the DiT backbone is shared across all physical steps.

*Remark* 3.1 (Surrogate loss as a learnable amortization). The residual-norm form and asymmetric split between $\theta$ and $(\theta_t, o_{t+1}, a_t)$ are motivated by the Bayes-derived construction in Section 2.3. This is a design connection, not a guarantee that the learned loss equals the explicit empirical or ideal loss. Proposition 4 shows that an unrestricted residual-gradient interface can express a continuous lifted update on a compact weight domain. Its construction requires access to the full weights and sufficiently expressive heads. The deployed fixed-probe heads have a smaller update class. Theorem 3.1 instead evaluates output beliefs under the training distribution and does not establish loss-level alignment, optimization convergence, or adaptation to an unseen likelihood.

[3]To highlight that they constitute the learnable part in the surrogate loss $\ell_\phi$ and are optimized together in $\mathcal{L}(\phi)$ (Section 3.2), we denote both $f_\phi$ and $g_\phi$ with the same subscript $\phi$ though they are separate neural networks.

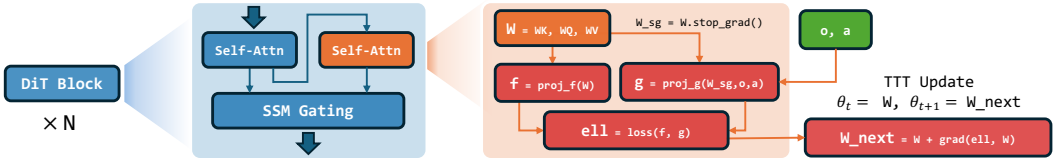



Figure 2: **Architecture overview of BFF.** A diffusion transformer (DiT) combines self-attention (Self-Attn), TTT modules, and state space model (SSM) gates. The updated weights $\theta$ (`W`) stack query, key, and value projections ($W_Q, W_K, W_V$). The surrogate loss $\ell_\phi$ uses learnable heads `proj_f` and `proj_g`. Details are in Appendix D.

### 3.2 Learning the Surrogate Loss

We train the update parameters $\phi$ on trajectories containing states, observations, and actions. The outer loss compares the predicted flow velocity with the difference between each recorded state and an independent Gaussian noise sample.

**Filtering operator error in probability space.** The two operators live in different spaces. $K$ acts on beliefs in $\mathcal{P}(\mathcal{X})$, while $\mathcal{K}_\phi$ acts on weights in $\Theta$. To compare them on equal footing, we use the weight-to-belief lift $\Phi : \Theta \to \mathcal{P}(\mathcal{X})$, $\theta \mapsto b_\theta$ from Section 2.2 (Equation (5)) and feed both operators the same current belief $\Phi(\theta_t)$, then measure the difference between their resulting next beliefs by

$$\mathcal{E}(\phi) \;:=\; \mathbb{E}_{t\sim U\{1:T-1\}}\, \mathbb{E}_{(h_t,\, o_{t+1},\, a_t)\sim P_{\text{data}}}\Big[W_2^2\Big(K\big(\Phi(\theta_t),\, o_{t+1},\, a_t\big),\; \Phi\big(\mathcal{K}_\phi(\theta_t,\, o_{t+1},\, a_t)\big)\Big)\Big],$$
$$\theta_t \;=\; \mathcal{K}_\phi^t(h_t) \;=\; \underbrace{\mathcal{K}_\phi(...\mathcal{K}_\phi(\mathcal{K}_\phi}_{\mathcal{K}_\phi \text{ applied } t \text{ times}}(\theta_0, o_1, a_0), o_2, a_1), ..., o_t, a_{t-1}) \tag{13}$$

where $T$ is the trajectory length, $U\{1{:}T-1\}$ samples physical time uniformly, and $h_t$ is the history of observations and actions from the dataset, $W_2$ measures the Wasserstein-2 distance between distributions, and $\theta_t$ is produced from $\theta_0$ by recursively applying $\mathcal{K}_\phi$ to the history $h_t \sim P_{\text{data}}$, identical to the filtering loop at test time. This error quantifies the distributional mismatch between the true Bayesian filtering update and the learned TTT update, averaged across the dataset distribution $P_{\text{data}}$. The pseudocode is in Algorithm 1.

**Training with state trajectories.** The operator error compares distributions and is generally unavailable during training. For trajectories drawn from the specified data-generating model, the conditional law of the recorded state $s_t$ given its history $h_t$ is $b_t$. We can therefore train the recursively generated belief using the conditional flow-matching loss below. Theorem 3.1 bounds the operator error by this population loss after subtracting its irreducible variance.

$$\mathcal{L}(\phi) \;=\; \mathbb{E}_{t\sim U\{1:T\}}\, \mathbb{E}_{\tau\sim U(0,1)}\, \mathbb{E}_{h_t,\, s_t,\, s_t^{(0)}}\Big[\big\|u_{\theta_t}(s_t^{(\tau)}, \tau) - (s_t - s_t^{(0)})\big\|^2\Big], \tag{14}$$

$\theta_t = \mathcal{K}_\phi^t(h_t)$ is obtained by iteratively applying $\mathcal{K}_\phi$ as in Equation (13), $s_t$ is the corresponding ground-truth state at step $t$, $s_t^{(0)} \sim \mathcal{N}(0, I)$ is a noise sample, and $s_t^{(\tau)} = \tau s_t + (1-\tau)s_t^{(0)}$ is the flow matching interpolant. We show next that $\mathcal{L}$ bounds the operator error in Equation (13) (proof in Appendix G).

**Theorem 3.1** (Training loss bounds the operator error)**.** *Assume finite second moments and well-defined flow ODEs, with each target marginal velocity generating its conditional interpolant marginals from the independent Gaussian source. Let $u_\theta$ be spatially Lipschitz with constant $L$, uniformly over weights, histories, and flow times. Let $K$ be $W_2$-Lipschitz with constant $L_K$ on the true and generated beliefs, uniformly over the observation-action pairs supplied to both belief arguments. Then for any horizon $T \geq 2$ there exist a constant $A > 0$ and a $\phi$-independent constant $C \geq 0$ such that the operator error in Equation* (13) *satisfies*

$$\mathcal{E}(\phi) \;\leq\; A \cdot \big(\mathcal{L}(\phi) - C\big), \qquad A \;=\; 4C_L\,(L_K^2 + 1), \tag{15}$$

*where $C_L$ depends only on $L$ and is the Grönwall factor in the appendix proof. The constant $C = T^{-1}\sum_{t=1}^{T} C_t$ is the time average of the irreducible conditional-flow-matching variances, averaged over data histories.*

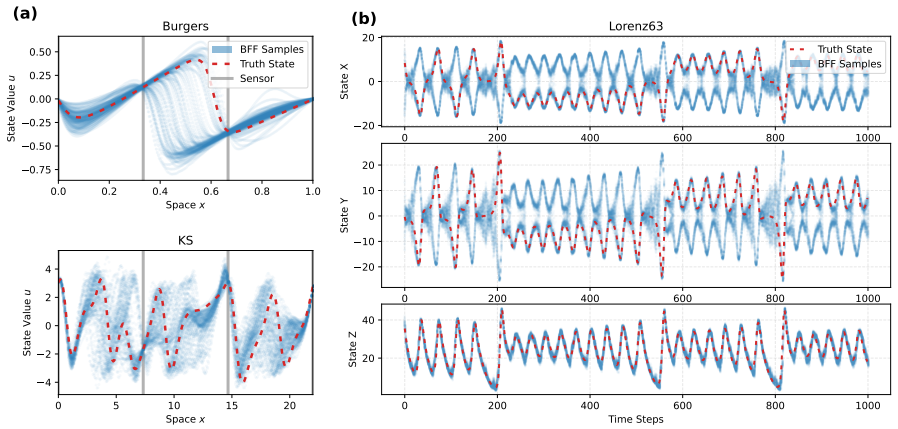



Figure 3: **Multimodal posterior samples.** **(a)** Burgers' and KS with four spatial sensors. **(b)** Lorenz-63 with observations of $Z$ only.

The bound implies that if $\mathcal{L}(\phi) - C = 0$ under the stated assumptions, then $\mathcal{E}(\phi) = 0$ over the modeled training distribution and horizon. For nonzero loss, the global Lipschitz and Grönwall factors can make $A$ loose, especially for sensitive dynamics. Numerical tightness has not been established. The statement applies to any recursive belief encoder trained with this outer loss and does not establish a theoretical advantage of a TTT update over a gated recurrent unit (GRU) update. It neither guarantees successful optimization nor controls unseen likelihoods or arbitrarily long rollouts.

## 4 Results

We test multimodal beliefs, reconstruction, and uncertainty under sparse or nonlinear observations on Lorenz-63, Burgers', KS, flow past three cylinders, and synthetic tokamak profiles. Baselines include EnKF, the extended Kalman filter (EKF), the ensemble score filter (EnSF), the Fourier neural operator (FNO), a Transformer, SDA, and a flow-based framework for data assimilation (FlowDAS). Dataset and baseline details are given in Appendices H and I.

### 4.1 Multimodal Posterior Representation

Figure 3 shows BFF posterior samples under sparse observations. Burgers' samples span multiple branches near the shock, while KS shows broad and sometimes multimodal distributions between sensors. These alternatives can be obscured by the posterior mean. In Lorenz-63, observing $Z$ leaves trajectories with simultaneously sign-reversed $X$ and $Y$ indistinguishable. BFF retains both branches, with one following the true trajectory. Further diagnostics and long-horizon results are in Appendices J.3 and K.

### 4.2 Distributional Accuracy of the Posterior

We evaluate KS, Burgers', and 2D NS under sparse, noisy observations. Table 1 reports relative $L_2$ error ($\mathcal{E}_{L_2}$), spectrum deviation ($\mathcal{E}_{\text{spec}}$), miscalibration area (MA), and marginal CRPS. BFF achieves the lowest $L_2$ error and CRPS on KS and Burgers' and ranks among the top two in all four metrics on NS. SDA has lower 1D MA, but higher reconstruction error and CRPS than BFF. On NS, SDA improves reconstruction error and CRPS, while BFF better preserves the spectrum and has lower MA. Thus BFF combines competitive reconstruction with spectral and calibration performance, although no method leads on every metric. Evaluation details are in Appendix H.5.

Figure 4 shows how these differences appear in the reconstructed fields, calibration curves, and spectra. EnKF has the lowest spectrum error on Burgers' and NS, but its NS intervals are poorly calibrated. Matching the distribution of energy across spatial scales therefore does not ensure that the predicted intervals cover the true state at the expected frequencies. Deterministic predictors have MA near one when evaluated as point masses.

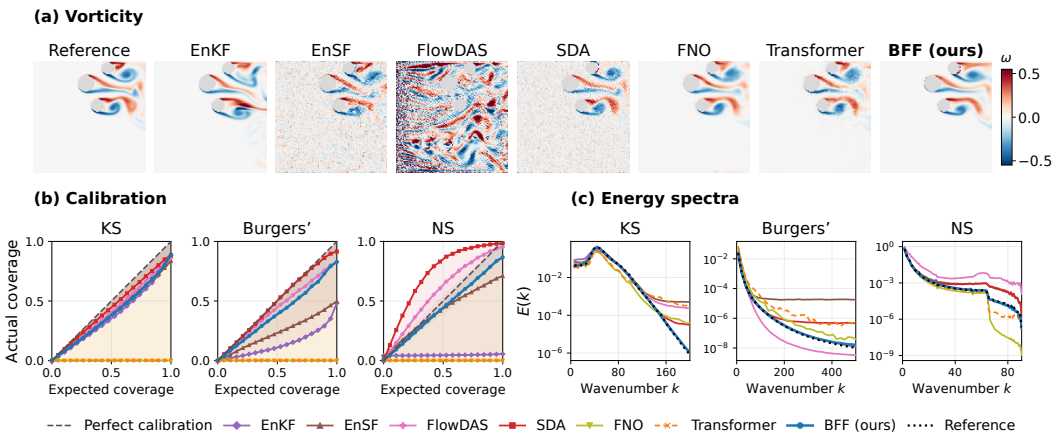



Figure 4: **Reconstruction, calibration, and spectra.** **(a)** Vorticity on one NS test trajectory with uniform sensors. **(b)** Actual versus expected coverage on the same 100 test trajectories as Table 1. The dashed diagonal denotes ideal calibration. **(c)** Mean sample spectra with the reference in black. KS and Burgers' use power spectra, and NS uses kinetic-energy spectra.

Replacing TTT with a GRU under the same backbone and training data increases NS tracking and spectrum errors (Appendix J.2). In a separate NS sweep, more probes improve MA but not reconstruction or CRPS (Appendix K, Table 8). Across five KS training seeds, the standard deviations of reconstruction error and CRPS are about $1\%$ and $2\%$ of their means, compared with about $10\%$ for MA (Appendix K). Calibration is therefore more sensitive to the training seed in this evaluation.

### 4.3 Realistic Applications

**Single moving sensor.** A single sensor follows a fixed zig-zag path through the NS flow domain (Figure 6). The filter reconstructs the $128 \times 128$ field from one measurement location per step, with the same path across test trajectories. Here, BFF has the lowest reported reconstruction error, spectrum deviation, and MA (Table 2). The declining entropy and ensemble spread in Appendix J.6 are consistent with observations progressively reducing spatial ambiguity about the obstacles and wakes.

**Tokamak profile estimation.** The synthetic EAST benchmark estimates radial temperature, poloidal-flux, and density profiles from electron cyclotron emission (ECE) and polarimeter-interferometer (POINT) measurements (Figure 5). The POINT observation map includes products of state components, making it nonlinear, while ECE has a linear readout with signal-dependent noise variance. The figure marks the plasma boundary as the last closed flux surface (LCFS). Physical models and data generation are detailed in Appendix H.4. In Table 2, BFF obtains the lowest gradient

Table 1: **Endpoint results on three PDE benchmarks.** Each reported row uses all 100 test trajectories and the same endpoint samples for relative $L_2$ error, spectrum deviation, miscalibration area (MA), and standard empirical CRPS. Lower is better. Deterministic predictors are treated as point-mass distributions. **Bold** and underline mark the lowest and second-lowest values.

| | 1D KS | | | | 1D Burgers' | | | | 2D NS (uniform sensors) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | $\mathcal{E}_{L_2}\downarrow$ | $\mathcal{E}_{\text{spec}}\downarrow$ | MA $\downarrow$ | CRPS $\downarrow$ | $\mathcal{E}_{L_2}\downarrow$ | $\mathcal{E}_{\text{spec}}\downarrow$ | MA $\downarrow$ | CRPS $\downarrow$ | $\mathcal{E}_{L_2}\downarrow$ | $\mathcal{E}_{\text{spec}}\downarrow$ | MA $\downarrow$ | CRPS $\downarrow$ |
| EnKF | 0.9802 | 0.6890 | 0.2425 | 1.1236 | 0.4974 | **0.0570** | 0.7140 | 0.0857 | 0.5339 | **0.0229** | 0.9367 | 0.3556 |
| EnSF | 1.0691 | 0.7408 | 0.2017 | 1.2860 | 0.9266 | 0.3669 | 0.5364 | 0.2014 | 0.6428 | 0.1628 | 0.3097 | 0.4035 |
| FlowDAS | 1.0105 | 0.7786 | 0.1746 | 1.1783 | 0.9996 | 0.1484 | 0.1746 | 0.2061 | 1.3065 | 0.3995 | 0.2475 | 0.6626 |
| SDA | 0.9345 | 0.6717 | **0.1242** | 1.0586 | 0.3074 | 0.2058 | **0.0817** | 0.0430 | **0.1608** | 0.1488 | 0.4313 | **0.0688** |
| FNO | 0.9756 | 0.8146 | 1.0000 | 1.4866 | 1.3790 | 0.1848 | 1.0000 | 0.2949 | 0.3281 | 0.2383 | 1.0000 | 0.1129 |
| Transformer | 0.9478 | 0.8052 | 1.0000 | 1.4337 | 1.3776 | 0.2467 | 1.0000 | 0.2946 | 0.3205 | 0.1276 | 1.0000 | 0.1086 |
| BFF (ours) | **0.8710** | **0.5549** | 0.2094 | **0.9588** | **0.2992** | 0.0816 | 0.2101 | **0.0417** | 0.2250 | 0.0286 | **0.1363** | 0.0747 |

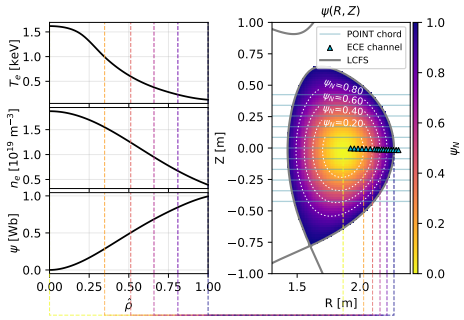



Figure 5: **Tokamak observation setup.** ECE and POINT measurements form a nonlinear observation model.

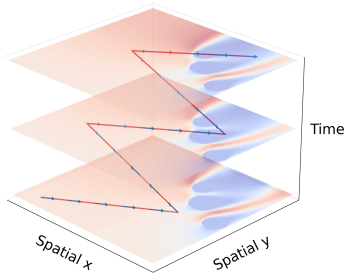



Figure 6: **Single-moving-sensor setup.** One sensor measures $(u, v, p)$ at each location along a prescribed zig-zag path.

Table 2: **Application results.** All tokamak methods share the MA protocol in Appendix H.4. Lower is better; bold/underline mark best/second best. Deterministic predictions are point masses.

| | | **Synthetic tokamak profiles** | | | | **NS, single moving sensor** | | |
|---|---|---|---|---|---|---|---|---|
| Method | sec./step ↓ | $\mathcal{E}_{L_2}$ ↓ | $\mathcal{E}_{\text{grad}}$ ↓ | $\mathcal{E}_{\text{spec}}$ ↓ | MA ↓ | $\mathcal{E}_{L_2}$ ↓ | $\mathcal{E}_{\text{spec}}$ ↓ | MA ↓ |
| EKF | 0.0499 | 0.1016 | 0.1657 | 0.2075 | 0.4275 | 0.5910 | 0.5844 | 0.8738 |
| EnKF | 0.0488 | 0.0870 | 0.1207 | 0.2039 | 0.9182 | 0.7859 | <u>0.0642</u> | 0.7328 |
| FNO | 0.0041 | <u>0.0174</u> | 0.0779 | 0.0688 | 1.0000 | 0.7812 | 0.1743 | 1.0000 |
| Transformer | 0.0031 | **0.0170** | 0.1140 | 0.0728 | 1.0000 | 0.7516 | 0.2766 | 1.0000 |
| SDA | 3.7466 | 0.2940 | 0.9483 | 0.0788 | 0.3682 | <u>0.5901</u> | 0.3133 | <u>0.5112</u> |
| FlowDAS | 0.8553 | 0.0299 | <u>0.0581</u> | <u>0.0629</u> | <u>0.2268</u> | 1.0733 | 0.2617 | 0.9456 |
| BFF (ours) | 0.0369 | 0.0193 | **0.0486** | **0.0101** | **0.0225** | **0.3771** | **0.0286** | **0.1799** |

error ($\mathcal{E}_{\text{grad}}$), spectrum error, and MA among the reported methods. The Transformer has lower relative $L_2$ error ($0.0170$ versus $0.0193$), but BFF has less than half its gradient error. BFF therefore recovers radial gradients more accurately despite the slightly larger error in profile values. Among the probabilistic methods, it also produces the best-calibrated marginal intervals.

### 4.4 Computation Efficiency

Fewer ODE steps reduce inference time and can slightly improve posterior-mean accuracy, as observed in the separate KS and NS step-count ablations (Appendix J.1). This speedup can come at the cost of distributional accuracy. On tokamak, reducing Euler integration from $50$ to five grid nodes preserves the reported mean error but increases MA. Solver choice matters too. Five midpoint nodes retain the same mean error and improve MA over $50$ Euler nodes while reducing latency from $225$ to $37$ ms. The TTT weight update alone takes $5.11$ ms.

## 5 Conclusion

We introduced BFF, a recursive generative filter that stores an evolving belief in flow-model weights and updates a small subset of those weights at test time. A Bayes-derived loss motivates the learned update, while the theory concerns an idealized interface and a conditional population-level error bound. BFF achieves the lowest reconstruction error and CRPS on KS and Burgers' and ranks among the top two methods in all four metrics on NS with uniform sensors. It also leads all three reported metrics with a single moving sensor. On synthetic tokamak profiles, BFF has lower gradient error and MA than the probabilistic baselines, while the Transformer has lower reconstruction error. Appendix K examines variability across training seeds and sensitivity to the update design and observation protocol. Together, these results show the value of retaining multiple plausible states while balancing reconstruction, spectrum, and calibration performance.

### AI USE STATEMENT

In this work, we used generative AI tools for the following tasks with required disclosure: help develop theoretical models or conceptual frameworks; formulate mathematical claims; propose or refine hypotheses; design or provide feedback on research methodology or experiments; implement methods; interpret results; provide critical ingredients for proving mathematical claims; assist in the writing of proofs; and assist with translation. We have not used generative AI tools for the following tasks with required disclosure: generate synthetic data sets; and clean and reformat datasets. Supporting qualitative and thematic data analysis does not apply to this work.

Additionally, we used generative AI tools for the following tasks with recommended disclosure: create or modify scientific figures or images; suggest experimental parameters; create or edit software code; creation of artifacts; draft parts of a research paper; summarize or analyze existing literature; discover research topics or identify gaps; brainstorming; sourcing/searching for information; and identify relevant literature. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

The reported experiments concern physical-system simulations and do not involve human participants or personal data. The tokamak benchmark uses synthetic trajectories and diagnostic signals based on magnetic geometry and reference profiles from EAST discharge 152444, as described in Appendix H.4. We evaluate state estimation without operating a physical device or testing a closed-loop controller. Use in physical control would require further validation of estimation errors, uncertainty calibration, and responses to filtering failures.

### REPRODUCIBILITY STATEMENT

An anonymous implementation is linked in Section 1. Appendix D describes the model architecture and training procedure, and Algorithms 1 and 2 specify the training and filtering loops. Appendix H documents the datasets, preprocessing, and evaluation metrics, while Appendix I describes the baseline implementations and parameter selection. The assumptions and proofs for the theoretical results are given in Appendices F and G. Sampling-cost ablations and additional evaluations across training seeds, update architectures, and observation protocols are reported in Appendices J and K.

## REFERENCES

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=li7qeBbCR1t.

Feng Bao, Zezhong Zhang, and Guannan Zhang. An ensemble score filter for tracking high-dimensional nonlinear dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, 432:117447, 2024.

Gal Berkooz, Philip Holmes, and John L Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics*, 25(1):539–575, 1993.

Siyi Chen, Yixuan Jia, Qing Qu, He Sun, and Jeffrey A Fessler. Flowdas: A flow-based framework for data assimilation. In *The Second Conference on Parsimony and Learning (Recent Spotlight Track)*, 2025.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023.

Youngseog Chung, Ian Char, Han Guo, Jeff Schneider, and Willie Neiswanger. Uncertainty toolbox: an open-source library for assessing, visualizing, and improving uncertainty quantification, 2021. URL https://arxiv.org/abs/2109.10254.

S.M. Cox and P.C. Matthews. Exponential time differencing for stiff systems. *Journal of Computational Physics*, 176(2):430–455, 2002.

Karan Dalal, Daniel Koceja, Jiarui Xu, Yue Zhao, Shihao Han, Ka Chun Cheung, Jan Kautz, Yejin Choi, Yu Sun, and Xiaolong Wang. One-minute video generation with test-time training. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 17702–17711, 2025.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Geir Evensen. Sequential data assimilation. In *Data Assimilation: The Ensemble Kalman Filter*, pp. 27–45. Springer, 2009a.

Geir Evensen. The ensemble Kalman filter for combined state and parameter estimation. *IEEE Control Systems*, 29(3):83–104, June 2009b.

F. Felici, O. Sauter, S. Coda, B. P. Duval, T. P. Goodman, J.-M. Moret, and J. I. Paley. Real-time physics-model-based simulation of the current density profile in tokamak plasmas. *Nuclear Fusion*, 51(8):083052, 2011. doi: 10.1088/0029-5515/51/8/083052.

Ruiqi Feng, Chenglei Yu, Wenhao Deng, Peiyan Hu, and Tailin Wu. On the guidance of flow matching. In *Forty-second International Conference on Machine Learning*, 2025.

Marco Fraccaro, Simon Kamronn, Ulrich Paquet, and Ole Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning. In *Advances in Neural Information Processing Systems*, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/7b7a53e239400a13bd6be6c91c4f6c4e-Abstract.html.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.13447.

Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pp. 107–113. IET, 1993.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Yinan Huang, Hans Hao-Hsun Hsu, Junran Wang, Bo Dai, and Pan Li. Accelerated sequential flow matching: A bayesian filtering perspective, 2026. URL https://arxiv.org/abs/2602.05319.

Rico Jonschkowski, Divyam Rastogi, and Oliver Brock. Differentiable particle filters: End-to-end learning with algorithmic priors. In *Robotics: Science and Systems*, 2018. URL https://arxiv.org/abs/1805.11122.

R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, March 1960.

Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt. Deep variational bayes filters: Unsupervised learning of state space models from raw data. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1605.06432.

Aly-Khan Kassam and Lloyd N. Trefethen. Fourth-order time-stepping for stiff pdes. *SIAM Journal on Scientific Computing*, 26(4):1214–1233, 2005.

Rahul G. Krishnan, Uri Shalit, and David Sontag. Deep kalman filters, 2015. URL https://arxiv.org/abs/1511.05121.

Randall J LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM, 2007.

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

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *11th International Conference on Learning Representations, ICLR 2023*, 2023.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2209.03003.

John Leask Lumley. The structure of inhomogeneous turbulent flows. *Atmospheric turbulence and radio wave propagation*, pp. 166–178, 1967.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182, October 2023. ISSN: 2380-7504.

François Rozet, Gérôme Andry, François Lanusse, and Gilles Louppe. Learning diffusion priors from observations by expectation maximization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 87647–87682. Curran Associates, Inc., 2024.

François Rozet and Gilles Louppe. Score-based Data Assimilation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 40521–40541, 2023.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9355–9366. PMLR, 18–24 Jul 2021.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (learn at test time): RNNs with expressive hidden states. In *Forty-second International Conference on Machine Learning*, 2025.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2302.00482.

Lloyd N Trefethen. *Spectral methods in MATLAB*. SIAM, 2000.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Gabriel D. Weymouth. Lily pad: Towards real-time interactive computational fluid dynamics, 2015. URL https://arxiv.org/abs/1510.06886.

G.D. Weymouth and Dick K.P. Yue. Boundary data immersion method for cartesian-grid simulations of fluid-body interaction problems. *Journal of Computational Physics*, 230(16):6233–6247, 2011.

Karen Willcox. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers & fluids*, 35(2):208–226, 2006.

X. Zhu, L. Zeng, H. Q. Liu, Y. X. Jie, Y. Yang, and X. Gao. Density profile evolution on EAST tokamak by the polarimeter/interferometer system. *Fusion Engineering and Design*, 131:29–32, 2018. doi: 10.1016/j.fusengdes.2018.04.036.

# A NOTATION

For ease of reading, Table 3 consolidates the symbols used throughout the paper. We follow the convention that *subscripts* (*for example*, $\theta_t, b_t$) index the discrete *physical time* of the filtering process, while *parenthesized superscripts* (*for example*, $s^{(\tau)}, p^{(\tau)}$) index the continuous *flow time* internal to the generative model.

Table 3: Summary of notation used in the main paper and the appendix.

| Symbol | Meaning |
|---|---|
| *Physical system and filtering problem* | |
| $s_t \in \mathcal{X} \subseteq \mathbb{R}^n$ | Physical state at time $t$ in the state space $\mathcal{X}$. |
| $o_t \in \mathcal{O} \subseteq \mathbb{R}^k$ | Observation at time $t$ in the observation space $\mathcal{O}$ (typically $k \ll n$). |
| $a_t \in \mathcal{A} \subseteq \mathbb{R}^m$ | Control action at time $t$ in the action space $\mathcal{A}$. |
| $h_t \coloneqq (o_{1:t}, a_{1:t-1})$ | History of past observations and actions up to time $t$. |
| $p(s_{t+1} \mid s_t, a_t)$ | Transition dynamics. |
| $p(o_t \mid s_t)$ | Observation model. |
| $b_t(s) \coloneqq p(s_t = s \mid h_t)$ | Belief state (posterior) at time $t$. |
| $\bar{b}_{t+1}$ | Prior obtained by applying the transition model to $b_t$ before the new observation. |
| $K : \mathcal{P}(\mathcal{X}) \times \mathcal{O} \times \mathcal{A} \to \mathcal{P}(\mathcal{X})$ | Recursive Bayesian filtering operator satisfying $b_{t+1} = K(b_t, o_{t+1}, a_t)$. |
| $T$ | Number of physical time steps in a trajectory. |
| *Flow matching belief representation* | |
| $\tau \in [0, 1]$ | Flow time internal to the generative model. |
| $s^{(\tau)}$, $p^{(\tau)}$ | Flow-time interpolant and its marginal distribution. |
| $u_\theta(s^{(\tau)}, \tau)$ | Velocity field of the flow matching model with parameters $\theta$. |
| $\phi^{u_\theta} : \mathcal{X} \to \mathcal{X}$ | Flow map induced by integrating $u_\theta$ from $\tau$=0 to $\tau$=1. |
| $\theta_t \in \Theta$ | Flow matching weights at physical time $t$ (the BFF belief representation). |
| $b_{\theta_t}$ | Belief distribution induced by $\theta_t$. |
| $\bar{b}_{\theta_{t+1}}$ | Prior obtained by applying the transition model to $b_{\theta_t}$ before the new observation. |
| $\Phi : \Theta \to \mathcal{P}(\mathcal{X}),\ \theta \mapsto b_\theta$ | Weight-to-belief lift. |
| *Filtering operator in weight space* | |
| $\mathcal{K}(\theta_t, o_{t+1}, a_t)$ | Ideal weight-space lift of $K$ when the target belief is representable. |
| $\mathcal{K}_\phi$ | Learned weight update used by BFF to approximate $\mathcal{K}$. |
| $\eta$ | Inner-loop (test-time) learning rate, learned jointly with $\phi$. |
| $\mathrm{sg}[\cdot]$ | Stop-gradient operator. |
| *Loss functions* | |
| $\ell^\star(\theta)$ | Ideal filtering loss, fitting $u_{Kb_{\theta_t}}$ under the propagated-prior path law (Equation (7)). |
| $\hat{\ell}(\theta)$ | Empirical filtering loss that estimates $\ell^\star$ up to a constant and finite-sample bias (Equation (9)). |
| $\ell_\phi(\theta; \theta_t, o_{t+1}, a_t)$ | BFF surrogate loss in latent space (Equation (11)). |
| $F_N(\theta)$, $G_N(\theta_t, o_{t+1}, a_t)$ | Stacked model and target velocity branches of the empirical filtering loss. |
| $f_\phi$, $g_\phi$ | Learned latent heads taking values in the latent space $\mathcal{H}$. |
| $\mathcal{H}$ | Shared latent space of $f_\phi, g_\phi$. |
| $\Pi_N : \mathbb{R}^{N \times n} \to \mathcal{H}$ | Latent map used in the latent amortization analysis. |
| $\mathcal{L}(\phi)$ | Outer-loop training loss (Equation (14)). |
| $\mathcal{E}(\phi)$ | Filtering operator error in $W_2^2$ (Equation (13)). |
| $N$, $N_j$ | Number of anchor samples and propagation Monte-Carlo samples. |
| $W_2$ | Wasserstein-2 distance. |

## B Related Works

**Non-learning filters.** Classical filters provide several representations of uncertainty (Kalman, 1960; Evensen, 2009b; Gordon et al., 1993). The extended Kalman filter (EKF) propagates a Gaussian approximation using local linearization. The EnKF propagates an ensemble and applies a covariance-based analysis. Its members need not be Gaussian, but the standard analysis does not generally recover a nonlinear, multimodal posterior. Particle filters use likelihood-weighted samples, with performance limited by weight degeneracy when observations are highly informative relative to the proposal. Variational approaches such as four-dimensional variational data assimilation (4D-Var) instead optimize a trajectory objective and require additional approximation or sampling to characterize posterior uncertainty. These methods trade representation, model access, and online cost differently from an amortized weight-space update.

**Diffusion- and flow matching posterior sampling-based filters.** The propagation and likelihood-guidance ingredients of the Bayes-derived velocity target are related to established guided diffusion and flow methods. The new element here is storing the persistent belief in online-updated generator weights. The linear interpolant also connects to rectified flow (Liu et al., 2023) and stochastic interpolants (Albergo & Vanden-Eijnden, 2023), in addition to standard flow matching (Lipman et al., 2023). Score-based data assimilation (Rozet et al., 2024) samples a joint state trajectory conditioned on observations. Repeating joint trajectory sampling during online assimilation incurs a cost that depends on the history window and sampling budget (Table 4). Likelihood-guided diffusion and flow models also require an approximation to the likelihood at intermediate sampling times (Chung et al., 2023; Feng et al., 2025). For example, with clean state $x_1$ and intermediate state $x_\tau$, the smoothed likelihood is

$$p(y \mid x_\tau) = \mathbb{E}_{x_1 \sim p(x_1 \mid x_\tau)}[p(y \mid x_1)], \tag{16}$$

and score guidance differentiates the logarithm of this expectation. Evaluating this conditional integral can be difficult in high dimensions. Practical approximations trade computational cost against guidance accuracy. Their error depends on the model and inference regime, so the empirical comparisons here do not establish a universal limitation of guided posterior samplers.

**Sequential transport filters.** Concurrent sequential flow matching (Huang et al., 2026) transports samples from the previous posterior to warm-start generation of the next one. BFF instead keeps a generator as the persistent belief and changes its weights through a learned gradient step. These approaches can be combined, for example by warm-starting samples from an updated BFF generator. We have not run a direct empirical comparison.

**Other learned recursive filters.** Deep Kalman filters (Krishnan et al., 2015), deep variational Bayes filters (Karl et al., 2017), and Kalman variational autoencoders (Fraccaro et al., 2017) learn latent state-space models and amortized inference components. Differentiable particle filters (Jonschkowski et al., 2018) retain weighted samples while training motion and measurement models end to end. BFF does not originate learned recursive filtering. It differs in using the online-updated parameters of a flow generator as the persistent belief representation. EnSF (Bao et al., 2024) is another relevant recursive baseline, evaluated in Table 1.

**Flow-based generative models and test-time training.** Diffusion models and flow matching learn generative dynamics along probability paths (Ho et al., 2020; Lipman et al., 2023). Flow matching trains a velocity field and generates samples by integrating its ODE. In BFF, this provides a differentiable generator whose selected weights can be updated as new observations arrive.

TTT (Sun et al., 2020; 2025) and fast-weight methods (Schlag et al., 2021) update selected parameters during inference. BFF uses these evolving parameters to encode a filtering belief. Its inner update minimizes a learned surrogate loss conditioned on the current weights, observation, and action, while the outer flow-matching objective trains the surrogate. TTT has also been applied to flow-based generative models (Dalal et al., 2025).

## C Limitations, Discussion and Future Work

**From existence to finite-capacity approximation.** Proposition 4 is an existence result for unrestricted heads that can access the full weights on a compact domain. It does not quantify the approximation error of the finite multilayer perceptron (MLP) projections or fixed-probe implementation. With one probe per online matrix, the gradient update has rank at most one along a fixed right factor. A 2D NS probe-count sweep showed no monotonic improvement across the tested configurations (Appendix K), but it cannot remove this structural limitation. The Bayes-derived losses motivate the surrogate's form. No training term forces the learned surrogate to match the ideal loss. The outer-loss theorem applies to any recursive belief encoder, including a GRU, and its nonzero-loss constant may be loose for chaotic systems.

**Amortization and online computation.** Amortization moves much of the filtering computation to offline training. The deployed weight update uses learned heads without querying a dynamics solver, differentiating an explicit likelihood, or evaluating the analytical Monte-Carlo correction. This is useful when diagnosis or control imposes a strict latency budget, as in the tokamak task. Posterior sampling still requires flow-ODE integration and dominates our measured inference cost (Appendix J.1). Explicit likelihood-guided updates can accommodate a supplied observation model at inference time, but their online numerical approximations also trade computation against posterior accuracy. The choice therefore depends on the application. Amortization favors repeated, fast inference within a specified or explicitly conditioned model family, while explicit model queries offer greater flexibility at an additional online cost.

**Observation-model conditioning and posterior diagnostics.** The moving-sensor update receives sensor coordinates together with the observation, so it explicitly conditions on the measurement location. Training on zigzag paths and testing unseen paths increases relative $L_2$ error from $0.463$ to between $0.578$ and $0.703$, showing sensitivity to observation geometry despite conditioning on sensor coordinates (Appendix K). The observation-noise model remains fixed during training and its parameters are not supplied at deployment. The noise-shift stress test therefore measures robustness, without establishing correct uncertainty adaptation to a new noise model. MA and CRPS assess scalar marginals. The PDE spectrum metric compares individual sample spectra, whereas the tokamak metric compares aggregate radial spectra. A bootstrap particle filter provides an approximate Lorenz reference. None alone validates the full joint posterior in high-dimensional tasks. The long-horizon and five-seed diagnostics cover their evaluated protocols only.

**Direct test-time optimization of the empirical loss.** The weight-space belief representation also permits a possible non-amortized extension that directly optimizes the empirical filtering loss $\hat{\ell}$ of Equation (9) at each observation. Given queryable dynamics and likelihood models, this would allow the target $G_N$ to incorporate the supplied models rather than their fixed training-time parameterization. It would require propagation samples, an explicit likelihood correction, and repeated optimization at inference time, trading the current update latency for model flexibility. The empirical objective estimates the ideal filtering loss under the propagated-prior path law. Its accuracy would depend on Monte-Carlo estimation, model capacity, and optimization. Sharply concentrated likelihoods can make the explicit correction expensive to estimate reliably (Appendix F.1). We leave implementation and evaluation of this extension to future work.

**Accelerated inference via more efficient flow samplers.** The midpoint solver with five grid nodes already brings per-step latency on the tokamak benchmark to $\sim 37$ ms (Appendix J.1), but ODE integration remains the dominant cost in our inference pipeline. Several generative-model objectives target fewer sampling evaluations. MeanFlow learns an average velocity over a flow-time interval for one-step generation (Geng et al., 2025). Rectified flow uses rectification to encourage straighter transport paths (Liu et al., 2023). Optimal-transport conditional flow matching uses transport couplings to simplify the learned flow (Tong et al., 2024). Integrating these objectives with the learned belief update could reduce posterior-sampling cost, but it would require new training and evaluation of filtering accuracy and calibration. We have not evaluated these extensions.

**Training overhead from second-order gradients.** The inner-loop second-order derivative used by the TTT update inflates the training-step cost by roughly $30\%$ over vanilla flow matching at

a comparable batch size (Appendix J.1). This overhead is moderate per step but does compound over long training runs. Reducing it via truncated backpropagation through the inner loop, implicit-differentiation tricks, or learned approximations of the weight update is a natural direction we have not pursued.

**Weight-as-feature for downstream tasks.** To alleviate particle-number limits on downstream tasks such as estimating statistical properties of the posterior, it is a promising direction to feed the TTT-updated weights themselves into downstream models that predict quantities of interest end-to-end. Our work demonstrates the feasibility of manipulating flow-based generative-model weights to tackle complex distributional operations in data assimilation, which opens such possibilities.

**Designing task-specific surrogate losses.** Proposition 4 motivates richer surrogate heads under its full-weight-access and expressivity assumptions. The deployed MLP or linear projections are a deliberate minimal choice, and their approximation gap remains unquantified. Incorporating physical structure or constraints directly into $\ell_\phi$ or the inner-loop optimization (for example, a conservation-aware projection after each TTT step) is a natural direction that we have not pursued here.

**Active, safety-aware sensor exploration.** Our base moving-sensor protocol uses a predefined zig-zag trajectory, and the additional path-shift tests also prescribe the sensor paths. These experiments leave open how to choose informative measurement paths online. Since observation paths heavily influence estimation accuracy in our experiments, coupling BFF with active-exploration algorithms that simultaneously maximize information gain and respect physical safety (for example, collision-free trajectories around obstacles) is a key open direction.

## D Implementation Details

This section unpacks the practical instantiation of BFF, walking through the backbone, the choice of online-updated weights $\theta$, the architecture of the surrogate-loss heads $f_\phi, g_\phi$, and the training pipeline. We organize the description around three concrete design questions, namely (i) which subset of the flow matching model constitutes $\theta$, (ii) how $f_\phi$ ingests the high-dimensional $\theta$, and (iii) how $g_\phi$ fuses the observation $o_{t+1}$ and action $a_t$.

**DiT backbone with per-layer TTT modules.** Our flow matching velocity field $u_\theta(s^{(\tau)}, \tau)$ is parameterized by a DiT (Peebles & Xie, 2023). Inside each transformer layer we insert a TTT module after self-attention and before a state space model (SSM)-style gate, so that the layer applies the sequence `Attention` $\rightarrow$ `TTT` $\rightarrow$ `SSMGating`. By default, the TTT module performs a second multi-head self-attention on top of the first attention's output, parameterized by a single per-layer weight $W^{(\ell)} \in \mathbb{R}^{3H\times H}$ that stacks the query, key, and value (QKV) projections of this second attention. The SSM-style gate then residually adds the TTT output back to the attention output. $W^{(\ell)}$ is the only online-updated parameter in the layer. It is evolved across physical time by a single gradient step on the inner surrogate loss (Equation (11)). All other DiT components (token embedding, the layer's first-attention QKV/output projections, MLP sub-layers, adaptive layer normalization (AdaLN) modulation, output head) form a static backbone shared across all physical steps. Figure 7 illustrates the integration and the per-step weight update.

**Online-updated weights $\theta$.** The online state $\theta$ comprises only the per-layer TTT weight matrices $W^{(\ell)} \in \mathbb{R}^{3H\times H}$, one per transformer layer (with $H$ the model hidden size). Each $W^{(\ell)}$ stacks the query, key, and value projections that drive the second self-attention computed on top of the layer's standard attention output. The full belief representation is the concatenation $\theta = \left(W^{(1)}, \ldots, W^{(L)}\right)$, where $L$ is the number of transformer layers. Each $W^{(\ell)}$ is updated independently of the others by the inner step, so the per-step gradient computation parallelizes across layers and depth. The static DiT backbone (token embedding, regular attention QKV/output, MLP sub-layers, AdaLN, output head) is *not* part of $\theta$ and is shared across all physical steps. It encodes the prior knowledge of the state distribution and dynamics learned during outer training. With our defaults ($H$=256, $L$=6), $\theta$ contains $\sim 1.2\times 10^6$ scalars. The inner loss updates these weights with one backward pass, while posterior generation through the flow ODE dominates inference time in the reported default configurations.

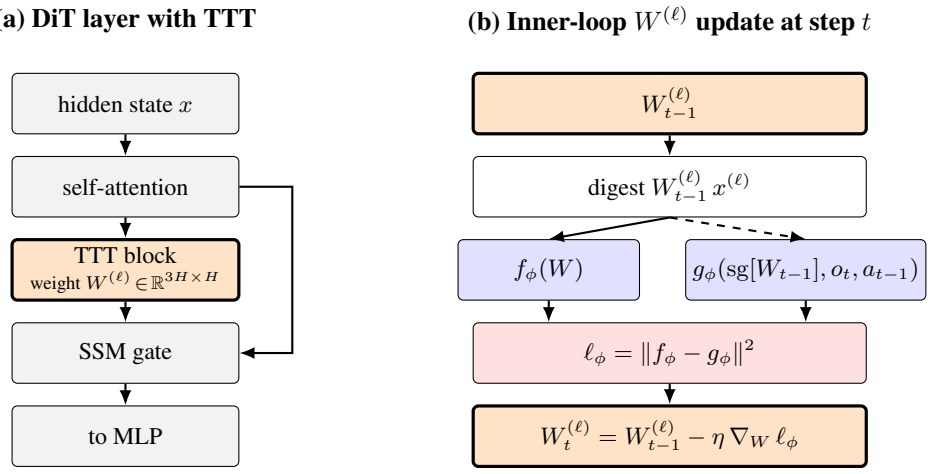



Figure 7: Architecture and inner-loop update of the TTT block. **(a)** Inside every transformer layer of the DiT, a TTT module sits between self-attention and an SSM-style residual gate. It performs a *second* multi-head self-attention on the first attention's output, parameterized by a single per-layer weight $W^{(\ell)} \in \mathbb{R}^{3H\times H}$ stacking $Q, K, V$. **(b)** At each physical step $t$, $W^{(\ell)}$ is updated by one gradient step on the inner surrogate loss $\ell_\phi$. Both heads $f_\phi = \texttt{theta_proj}(W \cdot x^{(\ell)})$ and $g_\phi = \texttt{target_proj}([\mathrm{sg}(W_{t-1}\cdot x^{(\ell)}), o_t, a_{t-1}])$ share the digest $W^{(\ell)}_{t-1}x^{(\ell)}$, but $g_\phi$ stop-gradients it (dashed arrow), realizing the asymmetric dependence on $\theta$ in Equation (11). Orange highlights the online-updated $W^{(\ell)}$. Gray boxes are the static backbone shared across all physical steps.

**Per-layer auxiliary parameters.** Three additional per-layer parameter groups parameterize the inner objective and are trained *only* by the outer loss.

- An initial weight $W^{(\ell)}_0 \in \mathbb{R}^{*\times H}$ from which the inner loop starts at the beginning of every trajectory.
- A scalar inner step size $\eta^{(\ell)} \in \mathbb{R}$, initialized to $0.01$ and learned jointly with $\phi$.
- A learnable probe vector $x^{(\ell)} \in \mathbb{R}^H$ that summarizes $W^{(\ell)}$ via the contraction $W^{(\ell)}x^{(\ell)}$ (see below).

We collect all three groups together with the heads $f_\phi, g_\phi$ defined next under the meta-parameter symbol $\phi$ used in the main text.

**Surrogate-loss heads $f_\phi$ and $g_\phi$.** A dense projection of every entry of $W^{(\ell)} \in \mathbb{R}^{*\times H}$ into a feature space would require a large projection layer. We instead summarize $W^{(\ell)}$ by its action on the learnable probe $x^{(\ell)}$, giving

$$\mathrm{pred}^{(\ell)} \;\coloneqq\; W^{(\ell)}x^{(\ell)} \;\in\; \mathbb{R}^{*}, \qquad * \in \{H,\, 3H\},$$

which is a smooth, low-dimensional digest of the full matrix that retains the linear dependence on $W^{(\ell)}$ needed for the inner gradient.

The model channel $f_\phi$ then projects $\mathrm{pred}^{(\ell)}$ into observation space as

$$f^{(\ell)}_\phi(\theta) \;=\; \texttt{theta_proj}^{(\ell)}\big(W^{(\ell)}x^{(\ell)}\big) \;\in\; \mathbb{R}^{\mathrm{obs_dim}},$$

where $\texttt{theta_proj}^{(\ell)}$ is a per-layer MLP (or, optionally, a linear map).

The target channel $g_\phi$ fuses the three inputs $(\mathrm{sg}[\theta], o_{t+1}, a_t)$ by raw feature-dimension concatenation followed by a single projection, giving

$$g^{(\ell)}_\phi(\mathrm{sg}[\theta], o_{t+1}, a_t) \;=\; \texttt{target_proj}^{(\ell)}\Big(\big[\,\mathrm{sg}\big(W^{(\ell)}x^{(\ell)}\big),\; o_{t+1},\; a_t\,\big]\Big) \;\in\; \mathbb{R}^{\mathrm{obs_dim}}.$$

Three points are worth noting. First, the heads are valued in observation space, so the latent space $\mathcal{H}$ in the main text is concretely $\mathbb{R}^{\text{obs_dim}}$. Second, the weight digest $W^{(\ell)}x^{(\ell)}$ entering $g$ is detached. The stop-gradient operator $\text{sg}[\cdot]$ blocks the inner-loop gradient from flowing into $g$'s side of the residual, so the inner update direction is shaped only by $f$'s $W^{(\ell)}$ path, while $g$'s own weights $\texttt{target_proj}^{(\ell)}$ remain in the outer-loop graph and are still trained by the outer flow-matching loss. Third, the target head combines the detached current-weight digest with the new observation and action through a single linear or MLP projection. The outer objective trains both heads through the resulting recursive belief updates.

**Inner update at inference.** Per layer, at every physical step, we perform the differentiable single gradient step (Algorithm 2)

$$W_{t+1}^{(\ell)} \;=\; W_t^{(\ell)} \;-\; \eta^{(\ell)}\, \nabla_{W_t^{(\ell)}} \left\| f_\phi^{(\ell)}(\theta_t) - g_\phi^{(\ell)}(\text{sg}[\theta_t], o_{t+1}, a_t) \right\|^2,$$

implemented by `torch.autograd.grad` with `create_graph=True` so that the inner update is itself differentiable for the outer-loop training described next. We use one inner step per observation and train the model with this update schedule. Proposition 4 establishes single-step expressivity for the idealized unrestricted interface, while Appendix K evaluates additional inference steps for the trained implementation.

**Training pipeline.** Outer training proceeds in two stages. **(1) Pretraining the static backbone.** We pretrain the DiT backbone together with $W_0$ on the marginal state distribution $p(s)$ with a standard flow-matching loss before outer training, which gives the inner loop a better starting point at small cost since no spatio-temporal dynamics are involved at this stage. **(2) Outer-loop training.** We then jointly train all parameter groups (DiT backbone, $W_0^{(\ell)}$, $\eta^{(\ell)}$, $x^{(\ell)}$, $f_\phi$, and $g_\phi$) with the unrolled outer-loop loss Equation (14). The bottleneck of this stage is the DiT forward pass through the unrolled trajectory rather than the lightweight TTT modules. We therefore subsample physical time steps when evaluating the flow-matching loss and initialize the gating with $\alpha = 0.1$ to stabilize early outer-loop dynamics. We optimize with adaptive moment estimation with decoupled weight decay (AdamW), using learning rate $10^{-4}$, weight decay $0$, gradient clip $1.0$, and a cosine schedule with $1{,}000$-step linear warmup. The inner step size $\eta^{(\ell)}$ uses a $10\times$ smaller learning rate to keep the inner step size stable in early training.

**Hyperparameters and efficiency.** Per-experiment hyperparameter choices (DiT depth/width, training horizon $T$, batch size, total outer steps) are reported in our released code, and sensor configurations and dataset specifics are reported in Appendix H. The tokamak timing diagnostic measures $5.11$ ms for the TTT update alone, compared with $36.92$ ms for the default five-node midpoint configuration including sampling (Table 5). In the measured large-batch regime, the additional training-step cost is approximately $30\%$ relative to vanilla flow matching (Figure 11).

# E BFF Algorithm Pseudocode

The pseudo-code for the training and inference loops of BFF are summarized in Algorithm 1 and Algorithm 2, respectively. Here $s_{[1:T]}$ denotes the state sequence from step 1 through $T$, with the same indexing convention for observations and actions. The training endpoint $s_t^{(1)}$ is the recorded state $s_t$, and $d\tau$ is the numerical integration step in flow time. The loss steps are the selected physical-time indices at which the training loss is evaluated.

# F Surrogate (Inner) Loss Design

This appendix provides the proofs of the lemmas and proposition stated in Section 2.3.

## F.1 Empirical Filtering Loss

This subsection proves the main-text Proposition 1 of the *Empirical filtering loss* paragraph. We state the score–velocity and conditional-flow identities in our notation. The linear interpolant is

**Algorithm 1** BFF Training.

**Input** Dataset $s_{[1:T]}, o_{[1:T]}, a_{[1:T-1]} \sim P_{\text{data}}$.
**Output** Trained $\phi, \theta_0, \eta$.
Pretrain $\theta_0$ on $s_t \sim P_{\text{data}},\ t \sim U\{1:T\}$ using standard flow matching loss
**for** $i = 0$ **to** train steps ▷ Outer Loop
  $\mathcal{L} \leftarrow 0$
  **for** $t = 1$ **to** $T$ **do** ▷ Inner Loop
    $\theta_t \leftarrow \theta_{t-1} - \eta \nabla_{\theta_{t-1}} \| f_\phi(\theta_{t-1}) - g_\phi(\text{sg}[\theta_{t-1}], o_t, a_{t-1}) \|^2$
    **if** $t \in$ loss steps **then**
      Sample $\tau \sim U(0,1),\ s_t^{(0)} \sim \mathcal{N}(0, I)$
      $s_t^{(\tau)} \leftarrow \tau s_t^{(1)} + (1-\tau) s_t^{(0)}$
      $\mathcal{L} \leftarrow \mathcal{L} + \| u_{\theta_t}(s_t^{(\tau)}, \tau) - (s_t^{(1)} - s_t^{(0)}) \|^2$
    **end if**
  **end for**
  Optimize $\phi, \theta_0, \eta$ with $\nabla_{\phi, \theta_0, \eta} \mathcal{L}$
**end for**

**Algorithm 2** BFF Inference.

**Input** Observation sequence $o_{[1:T]}, a_{[1:T-1]}$, trained model $\phi, \theta_0, \eta$.
**Output** Sampled states $\hat{s}_{[1:T]}$.
**for** $t = 1$ **to** $T$ **do** ▷ Physical Time Loop
  $\theta_t \leftarrow \theta_{t-1} - \eta \nabla_\theta \ell_\phi(\theta; \theta_{t-1}, o_t, a_{t-1}) \big|_{\theta = \theta_{t-1}}$
  Sample $s_t^{(0)} \sim \mathcal{N}(0, I)$
  **for** $\tau = 0$ **to** $1$ **step** $d\tau$ **do** ▷ Flow ODE Integration
    $s_t^{(\tau + d\tau)} \leftarrow s_t^{(\tau)} + u_{\theta_t}(s_t^{(\tau)}, \tau) \cdot d\tau$
  **end for**
  $\hat{s}_t \leftarrow s_t^{(1)}$
**end for**

shared with flow matching, rectified flow, and stochastic interpolants (Lipman et al., 2023; Liu et al., 2023; Albergo & Vanden-Eijnden, 2023). The three helper facts are a score–velocity affine bijection that is standard in flow matching, an additive split of the Bayesian update at the score level, and a conditional flow-matching identity that turns the propagation channel into an unbiased target.

**Why zero ideal loss recovers the posterior.** Fix $(\theta_t, o_{t+1}, a_t)$ and assume the updated posterior is well defined. For each $0 < \tau < 1$, convolving either endpoint distribution with the nondegenerate Gaussian path kernel gives a strictly positive density on $\mathbb{R}^n$. Thus the propagated-prior and updated-posterior path laws, with the same uniform flow-time law, are mutually absolutely continuous. Since the integrand in Equation (7) is nonnegative, $\ell^\star(\theta) = 0$ implies $u_\theta = u_{Kb_{\theta_t}}$ almost everywhere under both path laws. If the target velocity is realizable and the associated continuity equation has a unique solution represented by the flow ODE from the common Gaussian source, this equality yields the target posterior at the endpoint. This establishes the ideal update when a zero minimum is attained. For a restricted model class with nonzero residuals, the definition instead selects a best fit under the propagated-prior path law.

**Lemma F.1** (Score–velocity affine bijection)**.** *For the linear-Gaussian probability path $s^{(\tau)} = \tau s + (1-\tau) s^{(0)}$ with $s \sim q$ and $s^{(0)} \sim \mathcal{N}(0, I)$, the marginal velocity field $u_q(s, \tau) = \mathbb{E}[s' - s^{(0)} \mid s^{(\tau)} = s]$ and the marginal score $\nabla_s \log q^{(\tau)}(s)$ are related by*

$$u_q(s,\tau) \;=\; \frac{s}{\tau} + \frac{1-\tau}{\tau}\, \nabla_s \log q^{(\tau)}(s), \qquad \nabla_s \log q^{(\tau)}(s) \;=\; \frac{\tau\, u_q(s,\tau) - s}{1-\tau}, \tag{17}$$

*for any $\tau \in (0,1)$ and any $q \in \mathcal{P}(\mathcal{X})$ with differentiable marginal $q^{(\tau)}$.*

*Proof.* The path identity $s^{(\tau)} = \tau s' + (1-\tau)s^{(0)}$ rearranges to $s' - s^{(0)} = (s' - s^{(\tau)})/(1-\tau)$, hence taking conditional expectations gives

$$u_q(s,\tau) \;=\; \frac{\mathbb{E}[s' \mid s^{(\tau)} = s] - s}{1-\tau}. \tag{18}$$

The conditional density of $s^{(\tau)}$ given $s'$ is $\mathcal{N}(\tau s', (1-\tau)^2 I)$. We use the following form of Tweedie's formula (Efron, 2011). If $X = \mu + \sigma Z$ with $\mu \sim \pi$, $Z \sim \mathcal{N}(0, I)$ independent, and marginal density $m(x) = \int \mathcal{N}(x; \mu, \sigma^2 I)\, \pi(\mu)\, d\mu$, then

$$\mathbb{E}[\mu \mid X = x] \;=\; x + \sigma^2\, \nabla_x \log m(x). \tag{19}$$

We apply Equation (19) with $X = s^{(\tau)}$, $\mu = \tau s'$ (so the prior on $\mu$ is the pushforward of $q$ under $s' \mapsto \tau s'$, and the marginal of $X$ is exactly $q^{(\tau)}$), and $\sigma = 1 - \tau$. This yields

$$\mathbb{E}[\tau s' \mid s^{(\tau)} = s] \;=\; s + (1-\tau)^2\, \nabla_s \log q^{(\tau)}(s), \tag{20}$$

and dividing by $\tau$ gives

$$\mathbb{E}[s' \mid s^{(\tau)} = s] \;=\; \frac{s}{\tau} + \frac{(1-\tau)^2}{\tau}\, \nabla_s \log q^{(\tau)}(s). \tag{21}$$

Substituting Equation (21) into Equation (18) gives the first identity in Equation (17). The second is its inverse. □

The map Equation (17) is the technical bridge between scores and velocities under the flow-matching probability path. Every additive decomposition at the score level induces an additive decomposition at the velocity level with gain factor $\alpha(\tau) = (1-\tau)/\tau$.

**Lemma F.2** (Bayes split at the score level)**.** *Let $\pi \in \mathcal{P}(\mathcal{X})$ be a prior with differentiable density and $p(o \mid s)$ a likelihood with finite normalization $Z = \int p(o \mid s)\, \pi(s)\, ds \in (0, \infty)$. Define the Bayes posterior $\pi^o(s) := \pi(s)\, p(o \mid s)/Z$. At points where $\pi$ and $p(o \mid \cdot)$ are positive and differentiable,*

$$\nabla_s \log \pi^o(s) \;-\; \nabla_s \log \pi(s) \;=\; \nabla_s \log p(o \mid s). \tag{22}$$

*Proof.* Bayes' formula gives

$$\log \pi^o(s) - \log \pi(s) \;=\; \log p(o \mid s) - \log Z. \tag{23}$$

The normalization $\log Z$ does not depend on $s$, so differentiating in $s$ removes it and yields Equation (22). □

In the inner-loss derivation we apply Lemma F.2 to the prior–posterior pair $(\bar{b}_{\theta_{t+1}},\, K b_{\theta_t})$ defined relative to the model belief $b_{\theta_t}$, where $\bar{b}_{\theta_{t+1}}(s) := \int p(s \mid s', a_t)\, b_{\theta_t}(s')\, ds'$ is the propagated model prior and $K b_{\theta_t}$ is the Bayes update of $b_{\theta_t}$. The same identity Equation (3) applies with $b_t$ replaced by $b_{\theta_t}$.

**Lemma F.3** (Conditional flow-matching identity)**.** *Let $s' \sim b_{\theta_t}$, $s^p \sim p(\cdot \mid s', a_t)$, $s^{(0)} \sim \mathcal{N}(0, I)$, $\tau \sim U(0,1)$ (mutually independent given $s'$), and $s^{(\tau)} = \tau s^p + (1-\tau) s^{(0)}$ as in the anchor sampling of Section 2.3. Then*

$$\mathbb{E}\Big[ s^p - s^{(0)} \;\Big|\; s^{(\tau)} = s,\ \tau \Big] \;=\; u_{\bar{b}_{\theta_{t+1}}}(s, \tau). \tag{24}$$

*Proof.* We first identify the marginal law of the propagated anchor $s^p$, then match it to the linear-Gaussian probability path of Lemma F.1, and conclude by reading off the conditional mean.

Since $s' \sim b_{\theta_t}$ and $s^p \mid s' \sim p(\cdot \mid s', a_t)$, the marginal law of $s^p$, denoted by $\mathrm{Law}(s^p)$, is, by the tower property,

$$\mathrm{Law}(s^p) \;=\; \int p(\cdot \mid s', a_t)\, b_{\theta_t}(s')\, ds' \;=\; \bar{b}_{\theta_{t+1}},$$

the propagated model prior obtained by applying the prediction step Equation (1) to $b_{\theta_t}$ in place of $b_t$. Thus $s^p \sim \bar{b}_{\theta_{t+1}}$, and by construction $s^{(0)} \sim \mathcal{N}(0, I)$ is independent of $s^p$ (and of $\tau$). The triple

$(s^p, s^{(0)}, \tau)$ therefore realizes exactly the linear-Gaussian probability path of Lemma F.1 with data law $q = \bar{b}_{\theta_{t+1}}$, anchor noise $s^{(0)}$, and path point $s^{(\tau)} = \tau s^p + (1-\tau)s^{(0)}$.

Specializing Equation (18) to $q = \bar{b}_{\theta_{t+1}}$ gives

$$u_{\bar{b}_{\theta_{t+1}}}(s, \tau) \;=\; \frac{\mathbb{E}[s^p \mid s^{(\tau)} = s,\, \tau] - s}{1-\tau}.$$

On the other hand, the path identity $s^{(\tau)} = \tau s^p + (1-\tau)s^{(0)}$ rearranges as $s^p - s^{(0)} = (s^p - s^{(\tau)})/(1-\tau)$, so taking conditional expectation in $(s^{(\tau)}, \tau)$,

$$\mathbb{E}\Big[s^p - s^{(0)} \,\Big|\, s^{(\tau)} = s,\, \tau\Big] \;=\; \frac{\mathbb{E}[s^p \mid s^{(\tau)} = s,\, \tau] - s}{1-\tau}.$$

The two right-hand sides coincide, yielding Equation (24). □

**Anatomy of the empirical filtering loss.** We restate the constituents of Eqs. (9)–(10) with the dependencies the proof will use. For each anchor $i \in \{1, \dots, N\}$, draw a particle $s'_i \sim b_{\theta_t}$ from the current model belief, simulate one transition step $s_i^p \sim p(\cdot \mid s'_i, a_t)$, and pair it with independent draws $s_i^{(0)} \sim \mathcal{N}(0, I)$ and $\tau_i \sim U(0,1)$, giving the path point $s_i^{(\tau_i)} = \tau_i s_i^p + (1-\tau_i)s_i^{(0)}$. By the tower-property argument of Lemma F.3, the marginal law of $s_i^p$ is the propagated model prior $\bar{b}_{\theta_{t+1}}$, so $(s_i^p, s_i^{(0)}, \tau_i, s_i^{(\tau_i)})$ realizes the linear-Gaussian probability path of Lemma F.1 with data law $\bar{b}_{\theta_{t+1}}$.

The empirical filtering loss compares two branches at these path points. The model branch $F_N$ stacks $u_\theta(s_i^{(\tau_i)}, \tau_i)$, the standard flow-matching velocity to be fit. The target branch $G_N$ stacks a two-channel target. The first is the *propagation term* $s_i^p - s_i^{(0)}$, which by Lemma F.3 is conditionally unbiased for the prior velocity $u_{\bar{b}_{\theta_{t+1}}}(s_i^{(\tau_i)}, \tau_i)$. The second is the *likelihood term* $\alpha(\tau_i)\, \nabla_s \log \bar{p}^{(\tau_i)}(o_{t+1} \mid s)\big|_{s=s_i^{(\tau_i)}}$, with prefactor $\alpha(\tau) = (1-\tau)/\tau$, that converts the Bayesian score correction into a velocity contribution via the score-to-velocity affine map of Lemma F.1. The path-smoothed likelihood $\bar{p}^{(\tau)}$ is the central object that the proof will introduce in Equation (28) below. For numerical evaluation of the analytical loss, it can be replaced by the self-normalized Monte-Carlo estimator in Equation (31), using $N_j$ extra samples drawn from $\bar{b}_{\theta_{t+1}}$ independently of $s_i^p$.

We now state the full version of Proposition 1 and prove it.

**Proposition 2** (Empirical filtering loss, full version)**.** *Assume the posterior is well defined, the path scores exist, the self-normalized logarithmic gradient is almost surely well defined, and the conditional likelihood-target error satisfies the first-, second-, and fourth-moment bounds in Equation* (32)*. Assume a uniform finite fourth moment of the exact residual and a bounded first moment of the velocity's parameter derivative over the candidate-weight class, with differentiation permitted under the expectation. Use independent anchors and independent extra propagation samples for each anchor. Let $\ell^\star$ be Equation* (7)*, weighted by the same anchor law as $\hat{\ell}$. Then there exist constants $L_1, L_2, c_5 \geq 0$ and a $\theta$-independent $C$ such that*

$$\big|\mathbb{E}[\hat{\ell}(\theta)] - \ell^\star(\theta) - C\big| \;\leq\; \frac{L_1}{N_j}, \qquad \big\|\nabla_\theta \mathbb{E}[\hat{\ell}(\theta)] - \nabla_\theta \ell^\star(\theta)\big\| \;\leq\; \frac{L_2}{N_j},$$

*and*

$$\mathrm{Var}\big(\hat{\ell}(\theta)\big) \;\leq\; \frac{1}{N}\Big(\sigma^2 + \frac{c_5}{N_j}\Big),$$

*where $\sigma^2$ is the per-anchor residual variance under closed-form $\bar{p}^{(\tau)}$. The $1/N_j$ bias and variance contributions vanish identically whenever $\bar{p}^{(\tau)}$ admits a closed form (for example a linear-Gaussian observation model with a Gaussian prior).*

*Proof.* We first establish a propagation–likelihood decomposition of the target velocity at each flow time $\tau$, then check that the per-anchor stack of Equation (10) is conditionally unbiased for this target, and finally promote pointwise unbiasedness to a variance bound on the empirical loss.

**Bayesian velocity decomposition.** We derive the Bayesian score split directly for the smoothed marginals at $\tau \in (0,1)$, without requiring data-level densities. Let $\kappa_\tau(s \mid s') := \mathcal{N}(s; \tau s', (1-\tau)^2 I)$ be the conditional density of $s^{(\tau)}$ given the data anchor $s'$. Then

$$\bar{b}^{(\tau)}_{\theta_{t+1}}(s) = \int \bar{b}_{\theta_{t+1}}(s')\, \kappa_\tau(s \mid s')\, ds', \tag{25}$$

$$(Kb_{\theta_t})^{(\tau)}(s) = \frac{1}{Z_{t+1}} \int p(o_{t+1} \mid s')\, \bar{b}_{\theta_{t+1}}(s')\, \kappa_\tau(s \mid s')\, ds' = \bar{b}^{(\tau)}_{\theta_{t+1}}(s) \cdot \frac{\mathbb{E}_{\bar{b}^{(\tau)}_{\theta_{t+1}}(s' \mid s)}\big[p(o_{t+1} \mid s')\big]}{Z_{t+1}}, \tag{26}$$

where $\bar{b}^{(\tau)}_{\theta_{t+1}}(s' \mid s) := \bar{b}_{\theta_{t+1}}(s')\, \kappa_\tau(s \mid s') / \bar{b}^{(\tau)}_{\theta_{t+1}}(s)$ is the conditional law of the data anchor $s'$ given the path-level sample $s$ under the prior, and $Z_{t+1} = \int p(o_{t+1} \mid s')\, \bar{b}_{\theta_{t+1}}(s')\, ds'$ is the normalization of $Kb_{\theta_t}$. Taking $\nabla_s \log$ of the ratio in Equation (26) and noting that $\log Z_{t+1}$ does not depend on $s$,

$$\nabla_s \log (Kb_{\theta_t})^{(\tau)}(s) \;-\; \nabla_s \log \bar{b}^{(\tau)}_{\theta_{t+1}}(s) \;=\; \nabla_s \log \mathbb{E}_{\bar{b}^{(\tau)}_{\theta_{t+1}}(s' \mid s)}\big[p(o_{t+1} \mid s')\big]. \tag{27}$$

Equation (27) motivates the central object of this proof, the *path-smoothed likelihood* at flow time $\tau$, defined as

$$\bar{p}^{(\tau)}(o_{t+1} \mid s) \;:=\; \mathbb{E}_{\bar{b}^{(\tau)}_{\theta_{t+1}}(s' \mid s)}\big[p(o_{t+1} \mid s')\big] \;=\; \frac{\int p(o_{t+1} \mid s')\, \bar{b}_{\theta_{t+1}}(s')\, \kappa_\tau(s \mid s')\, ds'}{\int \bar{b}_{\theta_{t+1}}(s')\, \kappa_\tau(s \mid s')\, ds'}, \tag{28}$$

which averages the data-level likelihood against the kernel-weighted conditional law of the data anchor given the path point. With this definition, Equation (27) reads

$$\nabla_s \log (Kb_{\theta_t})^{(\tau)}(s) \;-\; \nabla_s \log \bar{b}^{(\tau)}_{\theta_{t+1}}(s) \;=\; \nabla_s \log \bar{p}^{(\tau)}(o_{t+1} \mid s), \tag{29}$$

and is exact at interior flow times whenever $0 < Z_{t+1} < \infty$ and the logarithmic scores are defined. At $\tau = 1$, conditioning on the data anchor gives $\bar{p}^{(1)}(o_{t+1} \mid s) = p(o_{t+1} \mid s)$ prior-almost everywhere. A data-level score identity additionally requires the positive differentiable densities of Lemma F.2. For $\tau < 1$, $\bar{p}^{(\tau)}$ is the likelihood on the path-level marginal $\bar{b}^{(\tau)}_{\theta_{t+1}}$. A Gaussian prior and a linear-Gaussian observation model give a closed-form path-smoothed likelihood, but it generally differs from the data-level $p(o_{t+1} \mid s)$ even in this case. Replacing it by the data-level likelihood is therefore generally an uncontrolled approximation, which is why we work with $\bar{p}^{(\tau)}$ throughout.

Applying Lemma F.1 separately to $q = Kb_{\theta_t}$ and $q = \bar{b}_{\theta_{t+1}}$ and subtracting (the deterministic $s/\tau$ term cancels) yields

$$u_{Kb_{\theta_t}}(s,\tau) \;-\; u_{\bar{b}_{\theta_{t+1}}}(s,\tau) \;=\; \frac{1-\tau}{\tau} \left[\nabla_s \log (Kb_{\theta_t})^{(\tau)}(s) - \nabla_s \log \bar{b}^{(\tau)}_{\theta_{t+1}}(s)\right].$$

Combining with Equation (29) gives the exact propagation–likelihood decomposition of the target velocity,

$$u_{Kb_{\theta_t}}(s,\tau) \;=\; u_{\bar{b}_{\theta_{t+1}}}(s,\tau) \;+\; \alpha(\tau)\, \nabla_s \log \bar{p}^{(\tau)}(o_{t+1} \mid s), \qquad \alpha(\tau) = \frac{1-\tau}{\tau}, \tag{30}$$

which holds at $\tau \in (0,1)$ for any observation model and propagated prior satisfying the above normalization and regularity conditions.

**Per-anchor unbiasedness.** Reading the anchor stack Equation (10) per coordinate, the per-anchor targets are $G_i^{\text{prop}} := s_i^p - s_i^{(0)}$ and

$$G_i^{\text{lik}} \;:=\; \alpha(\tau_i)\, \nabla_s \log \bar{p}^{(\tau_i)}(o_{t+1} \mid s)\big|_{s = s_i^{(\tau_i)}},$$

where $\bar{p}^{(\tau_i)}$ is the path-smoothed likelihood Equation (28). By Lemma F.3 the propagation target is conditionally unbiased for the prior velocity, $\mathbb{E}[G_i^{\text{prop}} \mid s_i^{(\tau_i)}, \tau_i] = u_{\bar{b}_{\theta_{t+1}}}(s_i^{(\tau_i)}, \tau_i)$. The likelihood target is a deterministic function of $s_i^{(\tau_i)}$ and $\tau_i$, and combining Equation (29) with Lemma F.1 gives

$$\mathbb{E}\Big[G_i^{\text{lik}} \,\Big|\, s_i^{(\tau_i)}, \tau_i\Big] \;=\; G_i^{\text{lik}} \;=\; u_{Kb_{\theta_t}}(s_i^{(\tau_i)}, \tau_i) - u_{\bar{b}_{\theta_{t+1}}}(s_i^{(\tau_i)}, \tau_i).$$

Adding the two expectations gives $\mathbb{E}[G_i^{\text{prop}} + G_i^{\text{lik}} \mid s_i^{(\tau_i)}, \tau_i] = u_{Kb_{\theta_t}}(s_i^{(\tau_i)}, \tau_i)$, that is the per-anchor target $G_i := G_i^{\text{prop}} + G_i^{\text{lik}}$ is conditionally unbiased for $u_{Kb_{\theta_t}}$.

**Constant-shifted unbiasedness of the closed-form loss and variance bound.** We first establish the result under exact $\bar{p}^{(\tau)}$, then quantify the $N_j$-dependent perturbation. Expanding the squared norm and taking conditional expectation in $G_i$ gives

$$\mathbb{E}\Big[\|u_\theta - G_i\|^2 \mid s_i^{(\tau_i)}, \tau_i\Big] \;=\; \big\|u_\theta - u_{Kb_{\theta_t}}\big\|^2 \;+\; \mathrm{Var}(G_i \mid s_i^{(\tau_i)}, \tau_i),$$

where Var of the vector target denotes its total conditional variance, the trace of its conditional covariance matrix. This term does not depend on $\theta$. Averaging over the propagated-prior anchor distribution gives $\mathbb{E}[\hat{\ell}(\theta)] = \ell^\star(\theta) + C$ with $C = \mathbb{E}[\mathrm{Var}(G_i \mid s_i^{(\tau_i)}, \tau_i)]$ a $\theta$-independent constant. In particular, $\nabla_\theta \mathbb{E}[\hat{\ell}(\theta)] = \nabla_\theta \ell^\star(\theta)$, and unrestricted pointwise minimization of the population quadratic gives $u^\star(s,\tau) = u_{Kb_{\theta_t}}(s,\tau)$ on the anchor support. Writing $F_i = u_\theta(s_i^{(\tau_i)}, \tau_i)$, the per-anchor summands $\|F_i - G_i\|^2$ are independent and identically distributed with variance at most $\sigma^2$, so $\mathrm{Var}(\hat{\ell}) \le \sigma^2/N$, which is the closed-form half of the variance bound.

**Monte-Carlo evaluation of the path-smoothed likelihood.** While Equation (28) defines $\bar{p}^{(\tau)}(o_{t+1} \mid s)$ as a ratio of two integrals over the propagated model prior $\bar{b}_{\theta_{t+1}}$, both integrals can be estimated from a shared set of extra propagation samples. For each anchor, draw $N_j$ independent samples $\{s_{i,j}^p\}_{j=1}^{N_j} \sim \bar{b}_{\theta_{t+1}}$, independently of its original propagation sample and of the samples used by other anchors. The self-normalized estimator

$$\hat{\bar{p}}^{(\tau_i)}(o_{t+1} \mid s) \;:=\; \frac{\sum_{j=1}^{N_j} \kappa_{\tau_i}(s \mid s_{i,j}^p)\, p(o_{t+1} \mid s_{i,j}^p)}{\sum_{j=1}^{N_j} \kappa_{\tau_i}(s \mid s_{i,j}^p)} \tag{31}$$

converges to $\bar{p}^{(\tau_i)}(o_{t+1} \mid s)$ almost surely as $N_j \to \infty$, and its $\nabla_s$ has a closed form through the kernel weights. Whenever $\bar{p}^{(\tau)}$ itself admits a closed form (for example a linear-Gaussian observation model with a Gaussian prior), Equation (31) is bypassed and the bounds below collapse to the closed-form result of the previous paragraph.

**Conditional moment assumptions for finite-$N_j$ bounds.** Differentiability and finite prior variance alone do not give uniform rates for the logarithmic gradient of the self-normalized estimator. We state the additional conditions used here explicitly. Write $A_i = (s_i^{(\tau_i)}, \tau_i)$ and $\varepsilon_i = \hat{G}_i^{\mathrm{lik}} - G_i^{\mathrm{lik}}$. Here $\hat{G}_i^{\mathrm{lik}}$ replaces $\bar{p}^{(\tau_i)}$ in $G_i^{\mathrm{lik}}$ by the estimator in Equation (31), and $\hat{G}_i = G_i^{\mathrm{prop}} + \hat{G}_i^{\mathrm{lik}}$ is the resulting total target. The kernel-weighted likelihood sum must be positive almost surely for the logarithmic gradient to exist. Assume, uniformly on the anchor domain and over the compact candidate-weight class,

$$\|\mathbb{E}[\varepsilon_i \mid A_i]\| \le c_3/N_j, \qquad \mathbb{E}[\|\varepsilon_i\|^2 \mid A_i] \le c_4/N_j, \qquad \mathbb{E}[\|\varepsilon_i\|^4 \mid A_i] \le c_6/N_j^2. \tag{32}$$

Also assume a uniform finite fourth moment for the exact residual $r_i = u_\theta(s_i^{(\tau_i)}, \tau_i) - G_i$, and a bounded first moment of $\|\nabla_\theta u_\theta\|$. The extra samples make $\varepsilon_i$ independent of $G_i^{\mathrm{prop}}$ conditional on $A_i$. These are sufficient moment hypotheses, not consequences proved here for arbitrary priors or likelihoods. Establishing them for a self-normalized score requires control of the kernel-weight denominators and their derivatives. Such control can fail in unbounded anchor tails even at fixed interior times, for concentrated likelihoods, or near $\tau = 1$, where the kernel collapses. The velocity prefactor also requires control near $\tau = 0$. Restricting flow time to an interior interval avoids the endpoints but does not establish the moment bounds, and restricts the population objective to that same interval. We do not claim that the finite-$N_j$ rates hold over the full uniform-time interval without the stated moment conditions.

**Loss and gradient bias.** Let $\beta_i = \mathbb{E}[\varepsilon_i \mid A_i]$ and define the deterministic conditional residual $\Delta_i = u_\theta(s_i^{(\tau_i)}, \tau_i) - u_{Kb_{\theta_t}}(s_i^{(\tau_i)}, \tau_i)$. Since $\mathbb{E}[G_i \mid A_i] = u_{Kb_{\theta_t}}(s_i^{(\tau_i)}, \tau_i)$,

$$\begin{aligned}\mathbb{E}[\|u_\theta - \hat{G}_i\|^2 \mid A_i] - \mathbb{E}[\|u_\theta - G_i\|^2 \mid A_i] \\ = -2\Delta_i^\top \beta_i + \mathbb{E}[\|\varepsilon_i\|^2 \mid A_i].\end{aligned}$$

The exact propagation variance cancels by conditional independence. If $\mathbb{E}\|\Delta_i\| \le D$, averaging over anchors gives

$$\big|\mathbb{E}[\hat{\ell}(\theta)] - \ell^\star(\theta) - C\big| \le \frac{2Dc_3 + c_4}{N_j} = \frac{L_1}{N_j}.$$

The anchor law and likelihood target are fixed when differentiating the candidate weights. With $\mathbb{E}\|\nabla_\theta u_\theta\| \le D'$ and differentiation under the expectation,

$$\|\nabla_\theta \mathbb{E}[\hat{\ell}] - \nabla_\theta \ell^\star\| \le 2D' c_3/N_j.$$

This bounds the gradient bias relative to the ideal filtering loss $\ell^\star$ under its propagated-prior path law.

**Variance of the average loss.** For one anchor write $Q_i = \|r_i\|^2$ and $B_i = \|r_i - \varepsilon_i\|^2 - \|r_i\|^2$. The conditional moment bounds imply $\mathbb{E}[B_i^2] = O(N_j^{-1})$, where $O(N_j^{-1})$ denotes a quantity bounded in magnitude by a constant times $N_j^{-1}$ as $N_j \to \infty$. Moreover,

$$|\mathbb{E}[B_i \mid A_i, G_i]| \le (2c_3\|r_i\| + c_4)/N_j.$$

The finite fourth moment of $r_i$ therefore gives $|\mathrm{Cov}(Q_i, B_i)| = O(N_j^{-1})$, where Cov denotes covariance. Consequently, $\mathrm{Var}(Q_i + B_i) \le \sigma^2 + c_5/N_j$ for suitable constants, where $\sigma^2$ bounds $\mathrm{Var}(Q_i)$. Independence across anchors yields

$$\mathrm{Var}(\hat{\ell}(\theta)) \le \frac{1}{N}\left(\sigma^2 + \frac{c_5}{N_j}\right).$$

Increasing $N$ reduces this variance. The finite-$N_j$ bias is controlled separately. None of these estimates is used during deployed amortized BFF updates. □

*Remark* F.1. The decomposition Equation (30) is exact at $\tau \in (0, 1)$ under the stated normalization and score regularity, using the path-smoothed likelihood of Equation (28). Neither the observation model nor the prior needs to be Gaussian. The endpoint likelihood identity holds prior-almost everywhere, but an endpoint score need not exist, for example for an atomic posterior. At $\tau < 1$, the path-smoothed likelihood generally differs from the data-level likelihood, including for a Gaussian prior and a linear-Gaussian observation model. The latter case permits a closed-form evaluation of the path-smoothed quantity, not its replacement by the data-level likelihood. The factor $\alpha(\tau) = (1-\tau)/\tau$ is dictated by the score–velocity affine map. Changing its relative weighting generally changes the target velocity and is not justified by the score identity.

**Explicit Monte-Carlo estimation and deployed amortization.** The sample budget $N_j$ belongs to the analytical estimator in Equation (31). Deployed BFF updates neither construct this estimator nor select $N_j$. A sharply concentrated likelihood can concentrate the weights of the explicit estimator on very few propagation samples, making its score estimate unreliable at a fixed sample budget. This numerical limitation concerns direct evaluation of $\hat{\ell}$. A separately trained amortized update can instead suffer an amortization shift when the deployed likelihood differs from its training family. Avoiding explicit Monte-Carlo weights does not guarantee robustness to that shift.

### F.2 Learned Latent Surrogate

This subsection proves the latent amortization proposition (Proposition 3) of the *Learned latent surrogate* paragraph. The proof bounds the deviation between a latent surrogate and the explicit empirical filtering loss in terms of the distortion of $\Pi_N$, the head approximation errors $\varepsilon_f, \varepsilon_g$, and the latent residual radius $B$. These are assumptions on the represented functions. The outer training loss does not directly enforce them.

**Proposition 3** (Latent amortization)**.** *For a fixed realization of the anchors and sampled targets, suppose there exists a linear latent map $\Pi_N : \mathbb{R}^{N\times n} \to \mathcal{H}$ that approximately preserves the stacked residual norm with distortion $\delta_N$, that the heads approximate $\Pi_N F_N$ and $\Pi_N G_N$ within errors $\varepsilon_f, \varepsilon_g$, and that the latent residual is bounded by $B$. Then*

$$\left|\ell_\phi(\theta; \theta_t, o_{t+1}, a_t) - \hat{\ell}(\theta)\right| \;\le\; \delta_N + 2B(\varepsilon_f + \varepsilon_g) + (\varepsilon_f + \varepsilon_g)^2. \tag{33}$$

*Proof.* Let

$$R_N(\theta; \theta_t, o_{t+1}, a_t) \;=\; F_N(\theta) - G_N(\theta_t, o_{t+1}, a_t) \tag{34}$$

denote the two-channel anchor residual on the compact domain $\mathcal{D}$ of tuples $(\theta, \theta_t, o_{t+1}, a_t)$ under consideration. The hypotheses on $\Pi_N$, $f_\phi$, $g_\phi$ spell out as

$$\left|\; \|\Pi_N R_N\|_{\mathcal{H}}^2 - \|R_N\|^2 \;\right| \;\le\; \delta_N, \qquad \|\Pi_N R_N\|_{\mathcal{H}} \;\le\; B, \tag{35}$$

and

$$\|f_\phi(\theta) - \Pi_N F_N(\theta)\|_{\mathcal{H}} \;\le\; \varepsilon_f, \tag{36}$$

$$\|g_\phi(\mathrm{sg}[\theta_t], o_{t+1}, a_t) - \Pi_N G_N(\theta_t, o_{t+1}, a_t)\|_{\mathcal{H}} \;\le\; \varepsilon_g. \tag{37}$$

Set $r_N = \Pi_N R_N$ and

$$e \;=\; [f_\phi(\theta) - \Pi_N F_N(\theta)] \;-\; [g_\phi(\mathrm{sg}[\theta_t], o_{t+1}, a_t) - \Pi_N G_N(\theta_t, o_{t+1}, a_t)].$$

By the triangle inequality and Eqs. (36)–(37), $\|e\|_{\mathcal{H}} \le \varepsilon_f + \varepsilon_g$. By linearity of $\Pi_N$, the BFF residual in $\mathcal{H}$ equals $r_N + e$, so

$$\left|\; \|r_N + e\|_{\mathcal{H}}^2 - \|r_N\|_{\mathcal{H}}^2 \;\right| \;\le\; 2\,\|r_N\|_{\mathcal{H}}\,\|e\|_{\mathcal{H}} + \|e\|_{\mathcal{H}}^2 \;\le\; 2B(\varepsilon_f + \varepsilon_g) + (\varepsilon_f + \varepsilon_g)^2,$$

using $\|r_N\|_{\mathcal{H}} \le B$ from Equation (35). Combining with the latent-distortion bound $\delta_N$ via the triangle inequality

$$\left|\ell_\phi - \hat{\ell}\right| \;\le\; \left|\ell_\phi - \|r_N\|_{\mathcal{H}}^2\right| + \left|\,\|r_N\|_{\mathcal{H}}^2 - \hat{\ell}\right|$$

yields Equation (33). □

*Remark* F.2. For a fixed realized anchor set, setting $\mathcal{H} = \mathbb{R}^{Nn}$, $\Pi_N = \mathrm{id}$ with id the identity map, $f_\phi = F_N$, and $g_\phi = G_N$ gives $\delta_N = \varepsilon_f = \varepsilon_g = 0$, hence $\ell_\phi = \hat{\ell}$. This is a conditional construction. Anchors vary with the current belief, action, and sampling, so a single head receiving only $\theta$ need not realize $F_N$ across all anchor sets. For finite learned heads, Proposition 3 is conditional on achieving the stated distortion and approximation errors. Increasing the latent dimension or head capacity may enlarge the available function class, but does not by itself ensure these errors decrease under the outer training objective.

### F.3 Expressivity of the Residual-Pullback Form

The result in this subsection concerns the unrestricted residual-gradient interface, not the finite fixed-probe implementation. It assumes that the heads can access the full weight vector and express the construction below on a compact domain. It does not bound the approximation error of the deployed linear/MLP heads or show that their learned loss agrees with the Bayes-derived loss.

**Proposition 4** (Residual-pullback expressivity)**.** *Let $\Delta^\star(\theta, o_{t+1}, a_t)$ be any continuous desired online update field on a compact domain in $\mathbb{R}^P$, where $P$ is the number of weight coordinates. If the target branch is allowed to depend on* $\mathrm{sg}[\theta]$*, then there exist heads $f^\star, g^\star$ such that a single gradient step on*

$$\|f^\star(\theta) - g^\star(\mathrm{sg}[\theta], o_{t+1}, a_t)\|^2 \tag{38}$$

*realizes*

$$\theta^+ = \theta + \Delta^\star(\theta, o_{t+1}, a_t). \tag{39}$$

*Proof.* Set

$$f^\star(\theta) = \theta, \qquad g^\star(\mathrm{sg}[\theta], o_{t+1}, a_t) = \mathrm{sg}[\theta] + \frac{1}{2\eta}\Delta^\star(\mathrm{sg}[\theta], o_{t+1}, a_t). \tag{40}$$

Because the second branch is stop-gradient with respect to $\theta$,

$$\nabla_\theta \|f^\star(\theta) - g^\star(\mathrm{sg}[\theta], o_{t+1}, a_t)\|^2 = -\eta^{-1}\Delta^\star(\theta, o_{t+1}, a_t). \tag{41}$$

One gradient step with step size $\eta$ therefore realizes Equation (39). □

Under standard universal approximation assumptions, neural heads can approximate the continuous maps in Equation (40) on compact domains. The exact construction is an expressivity result for the unrestricted residual-gradient interface. It does not quantify the approximation error of the deployed fixed-probe heads. Theorem 3.1 separately bounds population output error under its stated assumptions. It does not remove the representability, head-capacity, or optimization conditions needed to realize a Bayesian update.

## G Proof of Theorem 3.1 on the Filtering Operator Training Loss

We prove the population bound for histories drawn from the specified data-generating model. Write $b_t(\cdot \mid h_t)$ for its filtering posterior, and $\theta_t = \mathcal{K}_\phi^t(h_t)$ for the recursively generated weights. We assume finite second moments, well-defined flow ODEs, and the regularity needed for the marginal flow-matching velocity to generate the interpolant marginals. The spatial Lipschitz bound on the learned velocity is uniform over the weights, histories, and flow times under consideration. The filtering operator is Lipschitz in $W_2$ uniformly over the observation-action pairs under consideration, with the same pair supplied to both of its belief arguments.

*Proof.* **Conditional flow matching and the variance term.** Fix a physical step $t$ and history $h_t$. Draw $S_t \sim b_t(\cdot \mid h_t)$ and an independent $Z \sim \mathcal{N}(0, I)$, and define $X_\tau = \tau S_t + (1-\tau)Z$. The target marginal velocity is

$$v_t(x, \tau \mid h_t) = \mathbb{E}[S_t - Z \mid X_\tau = x, h_t]. \tag{42}$$

The conditional-to-marginal flow-matching identity (Lipman et al., 2023) gives

$$\begin{aligned}\mathcal{L}_t(\phi \mid h_t) &:= \mathbb{E}_{\tau, S_t, Z \mid h_t}\big[\|u_{\theta_t}(X_\tau, \tau) - (S_t - Z)\|^2\big] \\ &= R_t(\phi \mid h_t) + C_t(h_t), \end{aligned} \tag{43}$$

$$R_t(\phi \mid h_t) := \mathbb{E}_{\tau, X_\tau \mid h_t}\big[\|u_{\theta_t}(X_\tau, \tau) - v_t(X_\tau, \tau \mid h_t)\|^2\big], \tag{44}$$

$$C_t(h_t) := \mathbb{E}_{\tau, S_t, Z \mid h_t}\big[\|S_t - Z - v_t(X_\tau, \tau \mid h_t)\|^2\big]. \tag{45}$$

Here and below $\tau \sim U(0,1)$. The cross term vanishes by conditional expectation. The variance $C_t(h_t)$ depends on the posterior and may vary with both history and physical time, but is independent of $\phi$. Define

$$\mathcal{L}_t(\phi) = \mathbb{E}_{h_t}\mathcal{L}_t(\phi \mid h_t), \qquad C_t = \mathbb{E}_{h_t} C_t(h_t), \qquad C = \frac{1}{T}\sum_{t=1}^{T} C_t. \tag{46}$$

Then $\mathcal{L}_t(\phi) - C_t \geq 0$ and $\mathcal{L}(\phi) - C = T^{-1}\sum_{t=1}^{T}(\mathcal{L}_t(\phi) - C_t)$.

**The marginal loss bounds belief error.** For the fixed history, couple the target and learned flow ODEs from the same Gaussian draw as follows

$$\frac{dY_\tau}{d\tau} = v_t(Y_\tau, \tau \mid h_t), \qquad \frac{d\widehat{Y}_\tau}{d\tau} = u_{\theta_t}(\widehat{Y}_\tau, \tau), \qquad Y_0 = \widehat{Y}_0 = Z. \tag{47}$$

The marginal law of $Y_\tau$ is the law of $X_\tau$ conditional on $h_t$. Their individual paths need not coincide. For $e(\tau) = \|\widehat{Y}_\tau - Y_\tau\|^2$, the spatial Lipschitz bound and $2\langle a, b\rangle \leq \|a\|^2 + \|b\|^2$ give

$$\frac{de(\tau)}{d\tau} \leq (2L+1)e(\tau) + \|u_{\theta_t}(Y_\tau, \tau) - v_t(Y_\tau, \tau \mid h_t)\|^2. \tag{48}$$

Since $e(0) = 0$, Grönwall's inequality implies

$$e(1) \leq C_L \int_0^1 \|u_{\theta_t}(Y_\tau, \tau) - v_t(Y_\tau, \tau \mid h_t)\|^2 \, d\tau, \qquad C_L = e^{2L+1}. \tag{49}$$

The terminal pair is a coupling of $\Phi(\theta_t)$ and $b_t(\cdot \mid h_t)$. Taking expectation over $Z$ and using the target marginal path therefore yields

$$W_2^2\big(\Phi(\theta_t), b_t(\cdot \mid h_t)\big) \leq C_L R_t(\phi \mid h_t). \tag{50}$$

Averaging over histories, define the per-step belief error

$$D_t := \mathbb{E}_{h_t} W_2^2\big(\Phi(\theta_t), b_t(\cdot \mid h_t)\big) \leq C_L\big(\mathcal{L}_t(\phi) - C_t\big). \tag{51}$$

**Belief error bounds filtering-operator error.** For each realized history extension $(h_t, o_{t+1}, a_t)$, abbreviate the shared new-information arguments of $K$ and $\mathcal{K}_\phi$. Since $b_{t+1} = Kb_t$ and $\theta_{t+1} = \mathcal{K}_\phi(\theta_t)$, the triangle inequality gives

$$W_2\big(K\Phi(\theta_t), \Phi(\theta_{t+1})\big) \leq W_2\big(K\Phi(\theta_t), Kb_t\big) + W_2\big(b_{t+1}, \Phi(\theta_{t+1})\big). \tag{52}$$

The Lipschitz assumption on $K$ is

$$W_2\big(K\Phi(\theta_t), Kb_t\big) \le L_K W_2\big(\Phi(\theta_t), b_t\big). \tag{53}$$

Squaring and using $(a+b)^2 \le 2a^2 + 2b^2$ yields

$$W_2^2\big(K\Phi(\theta_t), \Phi(\theta_{t+1})\big) \le 2L_K^2 W_2^2\big(\Phi(\theta_t), b_t\big) + 2W_2^2\big(b_{t+1}, \Phi(\theta_{t+1})\big). \tag{54}$$

Taking expectation over the data history extension gives $2L_K^2 D_t + 2D_{t+1}$ on the right. By Equation (51),

$$\mathbb{E}W_2^2\big(K\Phi(\theta_t), \Phi(\theta_{t+1})\big) \le 2C_L L_K^2\big(\mathcal{L}_t(\phi) - C_t\big) + 2C_L\big(\mathcal{L}_{t+1}(\phi) - C_{t+1}\big). \tag{55}$$

The nonnegative per-step marginal losses imply

$$\frac{1}{T-1}\sum_{t=1}^{T-1}\big(\mathcal{L}_t(\phi) - C_t\big) \le \frac{1}{T-1}\sum_{t=1}^{T}\big(\mathcal{L}_t(\phi) - C_t\big) = \frac{T}{T-1}\big(\mathcal{L}(\phi) - C\big). \tag{56}$$

The same bound holds for the shifted sum over $t = 2, \dots, T$. Averaging Equation (55) over $t = 1, \dots, T-1$ and using $T/(T-1) \le 2$ for $T \ge 2$ gives

$$\mathcal{E}(\phi) \le 2C_L(L_K^2+1)\frac{T}{T-1}\big(\mathcal{L}(\phi) - C\big) \le A\big(\mathcal{L}(\phi) - C\big), \qquad A = 4C_L(L_K^2+1). \tag{57}$$

This proves Equation (15). A zero population marginal loss implies zero operator error over the modeled distribution and horizon. At nonzero loss, the global Lipschitz constants and Grönwall factor can make the bound loose. The proof does not require a gradient-based update. It applies to any deterministic recursive belief encoder with the stated properties, and does not establish optimization convergence or deployment outside the modeled distribution. □

## H Data Generation and Preprocessing

In this section, we elaborate on the data generation process for our benchmark systems. To evaluate the reconstruction performance across diverse physical regimes, we consider a series of one-dimensional (1D) and two-dimensional (2D) fluid systems. These benchmarks range from fundamental nonlinear PDEs to complex flows governed by the Navier-Stokes equations.

### H.1 1D Systems with Burgers' and Kuramoto-Sivashinsky Equations

Two representative 1D nonlinear partial differential equations (PDEs) are included, specifically selected to simulate shock formations and chaotic dynamics. These datasets are generated using high-fidelity numerical solvers, with boundary conditions tailored to each system to ensure physical consistency.

**Burgers' equation.** We utilize the viscous Burgers' equation,

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} + f, \tag{58}$$

to model the emergence and propagation of shocks, with viscosity $\nu = 0.01$. Physically, the nonlinear advection term drives the formation of steep gradients, while the diffusion term provides regularization.

We use a second-order finite difference scheme (LeVeque, 2007) for spatial discretization and the explicit Euler method for temporal integration. The system is solved on a spatial domain $x \in [0, 1]$ discretized into $N_x = 256$ grids. We simulate the system for $t \in [0, 1]$ with solver step $\Delta t_{\text{solver}} = 10^{-4}$. The stored state includes the initial condition and 100 subsequent snapshots at spacing 0.01 (101 states in total); the forcing has 100 saved interval values. The common endpoint evaluation retains state indices 0–99 and their corresponding forcing observations, so its endpoint is $t = 0.99$.

For the Burgers' equation, we enforce homogeneous Dirichlet boundary conditions, $u(0,t) = u(1,t) = 0$, by padding the field with zeros during each integration step. We generate initial conditions (ICs) to simulate shock wave formation and propagation. The initial condition is constructed as a superposition of two Gaussian pulses with randomized parameters,

$$u_0(x) = \sum_{i=1}^{2} A_i \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right). \tag{59}$$

We introduce a time-varying external forcing term $f(x,t)$. The force field is generated as a sum of spatiotemporal Gaussian blobs (Random Gaussian Field approximation),

$$f(x,t) = \sum_{j=1}^{M} A_j \exp\left(-\frac{(x-\mu_{x,j})^2}{2\sigma_{x,j}^2}\right) \exp\left(-\frac{(t-\mu_{t,j})^2}{2\sigma_{t,j}^2}\right), \tag{60}$$

where $M = 8$ is the number of potential components. Independently for each component, $\mu_{x,j} \sim \mathcal{U}[0, L]$, $\mu_{t,j} \sim \mathcal{U}[0, T]$, $\sigma_{x,j} \sim \mathcal{U}[0.05L, 0.15L]$, and $\sigma_{t,j} \sim \mathcal{U}[0.05T, 0.15T]$, with $L = T = 1$. In the convention of this equation, $A_1 \sim \mathcal{U}[-3, 3]$; for $j = 2, \ldots, 8$, $A_j = B_j \widetilde{A}_j$, where $B_j \sim \mathrm{Bernoulli}(1/2)$ and $\widetilde{A}_j \sim \mathcal{U}[-3, 3]$ independently. These amplitudes include the generator's factor of two multiplying its temporal Gaussian. Thus the first component is always active and each remaining component is independently switched off with probability $1/2$. The forcing varies across space, time, and trajectories.

For the initial condition (separately from the forcing coefficients above), we sample the parameters uniformly, with locations $\mu_1 \sim \mathcal{U}[0.2, 0.4]$, $\mu_2 \sim \mathcal{U}[0.6, 0.8]$, amplitudes $A_1 \sim \mathcal{U}[0, 2.0]$, $A_2 \sim \mathcal{U}[-2.0, 0]$, and widths $\sigma_i \sim \mathcal{U}[0.05, 0.15]$.

**Kuramoto-Sivashinsky (KS) equation.** We investigate the Kuramoto-Sivashinsky equation, a paradigmatic model for spatio-temporal chaos (Trefethen, 2000) in extended systems, governed by the fourth-order nonlinear PDE

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial x^2} + \frac{\partial^4 u}{\partial x^4} = 0. \tag{61}$$

This KS dataset is unforced ($f \equiv 0$); it does not use the Burgers' forcing process. The dynamics are characterized by a competition between a large-scale instability induced by the anti-diffusion term $u_{xx}$ and small-scale damping from the hyper-diffusion term $u_{xxxx}$. Ground truth trajectories are generated using a high-precision pseudo-spectral method combined with a stiffly-stable fourth-order Exponential Time Differencing Runge-Kutta (ETDRK4) (Cox & Matthews, 2002; Kassam & Trefethen, 2005) scheme. The system is discretized into $N_x = 256$ Fourier modes and integrated with a fine solver step of $\Delta t_{\text{solver}} = 5 \times 10^{-4}$ over a total duration of $T = 200$. To reduce the influence of initial transients, we discard the initial evolution phase ($t \in [0, 100]$) and retain the subsequent 100 time units for analysis. To vary the initial spatial profiles, initial conditions $u(x, 0)$ are sampled from a zero-mean Gaussian Process (GP). Crucially, we employ a periodic kernel to strictly enforce the boundary conditions,

$$k(x, x') = \sigma^2 \exp\left(-\frac{2\sin^2(\pi|x - x'|/L)}{\ell^2}\right), \tag{62}$$

where the kernel length-scale parameter is $\ell = 8.0$ and its amplitude parameter is $\sigma = 8.0$ (the kernel variance is $\sigma^2 = 64$). The final dataset comprises 12,000 trajectories (10,000 training, 2,000 testing), temporally subsampled at $\Delta t_{\text{out}} = 1.0$ to capture the long-term evolution of coherent structures.

**Sensor placement.** For each 1D system, we use four fixed sensors at stored grid indices 0, 85, 170, and 255. The Burgers' state contains 256 interior points, $x_j = (j+1)/257$ on $[0, 1]$. Its Dirichlet boundary values are imposed outside this stored state grid. The first and last sensors therefore lie near the boundaries, at $1/257$ and $256/257$, and are not identically zero. The KS grid is periodic with length $L = 64$ and points $x_j = 64j/256$, excluding the repeated endpoint at 64. Its sensors lie at 0, 21.25, 42.5, and 63.75, which are four distinct grid points. Both settings thus have a nominal state-to-sensor ratio of $256/4 = 64$. For Burgers', the shared observation bundle additionally contains the forcing at these same four locations. BFF, FNO, Transformer, and the selected SDA configuration

consume these sparse forcing observations; FlowDAS uses its state-only checkpoint. The numerical filters maintain a latent forcing trajectory drawn jointly with the initial state from training data and update it from sparse forcing measurements. No method receives the complete held-out forcing field. KS has neither forcing observations nor a latent forcing input. Observations are sampled at the dataset's snapshot interval. The sensor placement and example trajectories are shown in Figures 9b, 9a, 10a, and 10b.

### H.2 2D Navier-Stokes Flow Past Multiple Cylinders

We consider incompressible flow past three stationary cylinders. The state contains the two velocity components and pressure on a $128 \times 128$ grid. Cylinder locations and sizes vary between trajectories. The saved data provide each trajectory's three boundary contours and Reynolds number. The governing equations are

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{u}, \tag{63}$$

$$\nabla \cdot \mathbf{u} = 0. \tag{64}$$

The cylinder geometry changes the wake structure and makes reconstruction from sparse observations more difficult. We do not assume that the three wakes evolve independently.

The dataset was generated with `Lily Pad` (Weymouth, 2015), which uses the Boundary Data Immersion Method (BDIM) (Weymouth & Yue, 2011). The available files contain $9{,}974$ training trajectories and $1{,}993$ test trajectories, with $50$ saved frames per trajectory. Our main comparison uses the first $100$ test trajectories in stored order. The files do not record frame timestamps or an exact generator version, so they do not establish the original saving interval. For the numerical filtering baselines, we use a BDIM reference step of $0.01$ and select ten solver steps per observation interval using training trajectories. The resulting interval of $0.1$ is a reconstruction choice rather than recovered timestamp metadata. These are known-parameter numerical references. Each filter receives the target cylinder geometry and Reynolds number and uses that geometry when initializing its ensemble. The state representation and subsequent constraints are specified for each filter below. The neural methods do not receive these parameters.

We retain the preprocessing used by the existing NS checkpoints. Channel normalization uses the statistics of the corresponding dataset split, with a $10^{-6}$ stabilizer in the standard deviation, and the target fields are masked inside the cylinders. Using test-split statistics is a transductive preprocessing limitation of this benchmark. All methods are scored in these normalized units over the same full target grid. Neural predictions are not given an additional mask derived from the test geometry.

**Sensor placement.** We consider two complementary observation regimes for the 2D Navier-Stokes system. The first is a structured sparse setting based on a coarse-grained uniform grid. We downsample the high-resolution spatial domain ($\Omega \in \mathbb{R}^{128\times128}$) by a factor of $scale = 16$, placing sensors at the centroids of non-overlapping $16 \times 16$ local patches (with a half-interval spatial offset). This yields $N_{obs} = (128/16)^2 = 64$ stationary measurement points that observe all three state channels and provide a regular but low-resolution view of the flow dynamics. An illustrative snapshot of the sensor layout is shown in Figure 8.

The second is a *single-moving-sensor* observation mode, designed to test extreme data compression. At each physical time step, one mobile sensor measures all three channels $(u, v, p)$ at one spatial location, giving a three-component observation. The full state has $3 \times 128 \times 128$ scalar entries, so the nominal state-to-measurement dimension ratio remains $(3 \times 128 \times 128)/3 = 16384 : 1$. This ratio counts measured field components; the prescribed sensor coordinates are supplied separately. The sensor follows a deterministic trajectory combining a horizontal zig-zag sweep (20-frame period) and a linear vertical descent over the total duration $T$, bounded by a $10\%$ spatial margin. To ensure fair and consistent evaluation, this identical predefined trajectory is rigidly applied across all test samples. Consequently, we assume an idealized sensor without collision avoidance constraints. The joint optimization of active exploration efficiency and physical safety is left for future work. The scan path and quantitative results are presented in Figure 6 and Table 2, with additional visualizations in Appendix J.

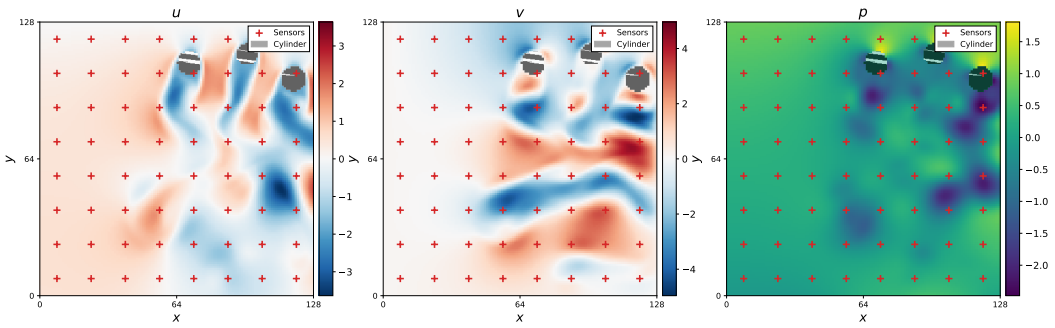
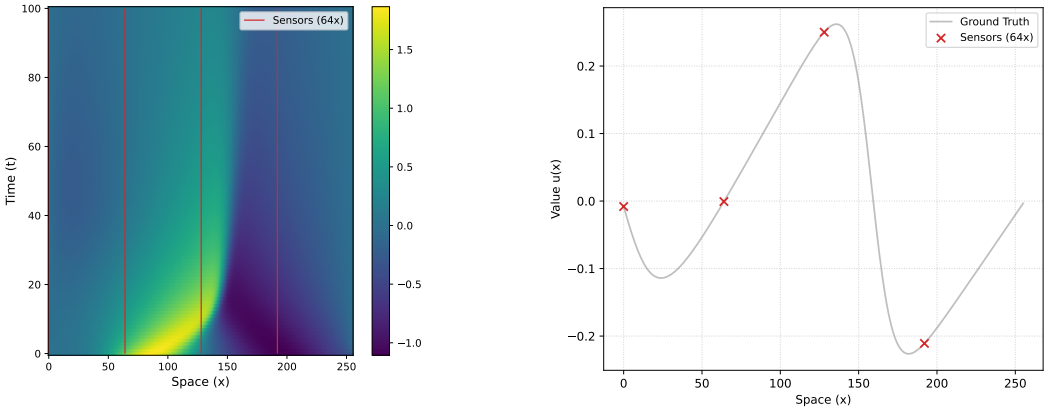
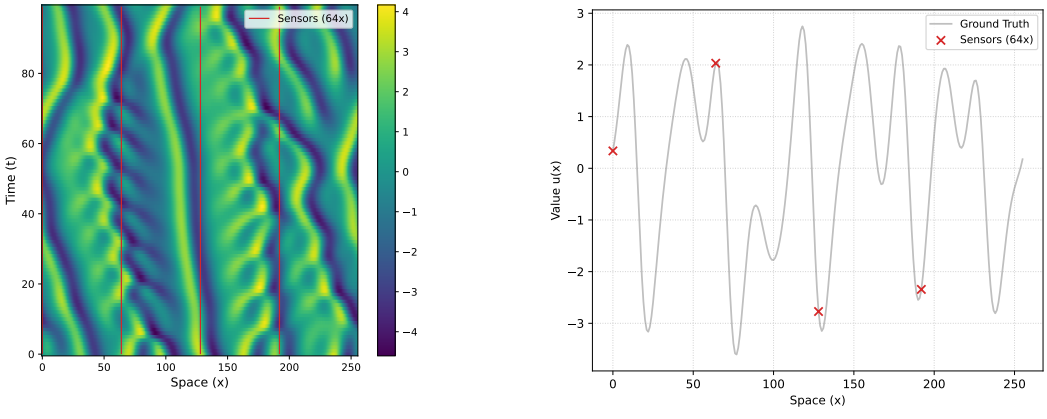



Figure 8: Illustration of the Navier-Stokes uniform $8 \times 8$ sensor placement and an example state.




(a) Entire trajectory. (b) Single snapshot.

Figure 9: Illustration of the Burgers' equation in the dataset. The four sensors are highlighted in red.




(a) Entire trajectory. (b) Single snapshot.

Figure 10: Illustration of the KS equation in the dataset. The four sensors are highlighted in red.

### H.3 LORENZ-63 SYSTEM

The Lorenz-63 system is a classic three-dimensional chaotic ODE,

$$\dot{X} = \sigma(Y - X), \qquad \dot{Y} = X(\rho - Z) - Y, \qquad \dot{Z} = XY - \beta Z, \tag{65}$$

with the standard parameters $(\sigma, \rho, \beta) = (10, 28, 8/3)$, for which the system exhibits a butterfly-shaped strange attractor. We use it as a stress test for long-horizon recursive filtering in which only the $Z$ coordinate is observed (with additive Gaussian noise, $\sigma_o = 0.5$), while $X$ and $Y$ remain hidden. The Lorenz dynamics and $Z$ observations are unchanged by $(X, Y, Z) \mapsto (-X, -Y, Z)$. With our symmetric initial distribution, sign-reversed trajectories therefore have equal posterior weight. The filter must preserve this ambiguity as it tracks the hidden $X$ and $Y$ coordinates.

**Dataset.** Trajectories are integrated with an explicit fourth-order Runge-Kutta scheme at $\Delta t = 0.01$. Each trajectory begins with a 20 time-unit burn-in from an initial state drawn uniformly within $[-20, 20]^3$ to reduce the influence of initial transients. The dataset consists of $10^4$ training trajectories, each of length $T_{\text{train}} = 100$, yielding $\approx 1.2 \times 10^5$ lobe-switching events in total. Test trajectories are generated under the same protocol but are extended to $T_{\text{test}} \in \{1000, 4000\}$, that is, $10\times$ and $40\times$ the training horizon, to evaluate generalization to far longer recursive inference.

### H.4 TOKAMAK PROFILE ESTIMATION

We construct a synthetic filtering benchmark using the magnetic geometry and prepared reference profiles for EAST discharge 152444. The task is to recover radial plasma profiles from local temperature measurements, chord-integrated measurements, and known heating commands. Profile estimation of this kind supports model-based plasma control (Felici et al., 2011). Here, the trajectories and diagnostic signals are generated by a simulator.

**State and geometry.** The state contains electron temperature $T_e$, axis-referenced poloidal flux $\psi(\hat{\rho}) - \psi(0)$, and electron density $n_e$ on a grid of 64 radial points. The transport coordinate $\hat{\rho} = \sqrt{\Phi/\Phi_b} \in [0, 1]$ labels normalized toroidal flux, with 0 at the magnetic axis and 1 at the reference plasma boundary. The Equilibrium Fitting (EFIT) reconstruction of discharge 152444 supplies the background geometry and the map between these radial labels and the $(R, Z)$ plane. This geometry is held fixed during each trajectory. The three profiles form the reduced state of the transport model.

**Dynamics and data generation.** The simulator advances $T_e$ and $\psi$ with a coupled heat-transport and poloidal-flux-diffusion solver, using its `felici_ff` heat-transport closure and implicit Euler integration. Density is randomized between trajectories and held fixed within each trajectory. Each simulation runs for 0.1 s with a 1 ms integration step. Saving the initial state and every fifth solver step gives 21 frames at times $0, 5, \ldots, 100$ ms. The dataset contains 10,000 training trajectories and 1,000 test trajectories.

Initial temperature and density profiles are varied around the reference profiles through independently sampled core gains, radial widths, and edge gains. Their uniform sampling ranges are $([0.50, 1.15], [0.32, 0.80], [0.75, 1.05])$ for $T_e$ and $([0.70, 1.30], [0.35, 0.90], [0.85, 1.15])$ for $n_e$, in that order. The action has 12 components, comprising deposition radius, deposition width, and injected power for each of four electron-cyclotron-heating channels. Commands are piecewise constant, with radius in $[0.05, 0.85]$, width in $[0.12, 0.28]$, power in $[0.2, 0.9]$ MW, and segment duration in $[25, 80]$ ms. All four channels share a start time sampled uniformly from $[0, 100]$ ms. In each subsequent segment, each channel is independently switched off with probability 0.1. If all four are off, one is chosen at random and set to the minimum power. The filter receives the action history together with the observations.

**Synthetic diagnostics.** The observation vector has 36 components. An electron-cyclotron-emission (ECE) model samples $T_e$ at 14 retained radial locations from the discharge data. The other 22 components are interferometric and Faraday-rotation signals on 11 horizontal chords, following the arrangement of the EAST POINT system (Zhu et al., 2018). Chord heights run from $-0.425$ to

$0.425\,\mathrm{m}$ in increments of $0.085\,\mathrm{m}$, and each line integral uses $512$ points in $R$,

$$\phi_j = C_\phi \int_{\ell_j} n_e \,\mathrm{d}\ell, \qquad \psi_{F,j} = C_F \int_{\ell_j} n_e B_\parallel \,\mathrm{d}\ell, \tag{66}$$

with simulation scale factors $C_\phi = 2 \times 10^{-19}\,\mathrm{deg\,m^2}$ and $C_F = 10^{-19}\,\mathrm{deg\,m^2\,T^{-1}}$. The fixed geometry maps each profile onto the chords, and spatial derivatives of $\psi$ determine the poloidal-field contribution to $B_\parallel$. Thus the interferometric signal is linear in $n_e$, whereas the Faraday signal couples density and magnetic flux through their product. Line integration reduces spatial resolution. The product coupling supplies the nonlinear part of this observation model.

**Noise and normalization.** ECE noise is Gaussian with standard deviation

$$\sigma_{\mathrm{ECE}}(y_{\mathrm{clean}}) = \sqrt{(20\,\mathrm{eV})^2 + (0.03\,|y_{\mathrm{clean}}|)^2}. \tag{67}$$

This makes the noise variance depend on the signal while leaving the noiseless temperature readout linear. Each interferometric chord has independent noise with standard deviation $1.0°$, plus a shared $0.35°$ Gaussian component across the chord bundle. The Faraday signals have independent Gaussian noise with standard deviation $0.1°$. The shared component therefore correlates the interferometric errors across chords. Normalization statistics are estimated over trajectories and time using up to $1{,}000$ sampled training trajectories. States are normalized separately at each channel and radial point. Observations and actions are normalized by component.

**Evaluation window and coverage.** Profile-error and radial-spectrum comparisons use the first $32$ test trajectories and all $21$ frames. The default BFF configuration uses five midpoint grid nodes and $100$ posterior samples per trajectory. For every method in the tokamak columns of Table 2, MA uses the same first $32$ test trajectories, excludes the first five saved frames, and retains the $16$ frames from $25$ to $100\,\mathrm{ms}$. All methods use the same central-interval definition, quantile levels, linear interpolation, and pooling order: coverage is pooled across trajectories, retained frames, and all $192$ state coordinates before taking the absolute deviation from nominal coverage. Deterministic methods are evaluated as point-mass predictive distributions under the same rule. With $q_\ell = \ell/19$ and linearly interpolated empirical quantiles, this gives

$$\mathrm{MA}_{\mathrm{pool}} = \frac{2}{20} \sum_{\ell=0}^{19} \left| \frac{1}{32 \cdot 16 \cdot 192} \sum_{n,t,j} \mathbf{1}\{Q_{n,t,(1-q_\ell)/2,j} \le s^*_{n,t,j} \le Q_{n,t,(1+q_\ell)/2,j}\} - q_\ell \right|. \tag{68}$$

The same formula is used for all methods in the tokamak comparison; it also reproduces the default BFF MA from its saved predictions. Pooling before the absolute value can hide differences between coordinates. It differs from the coordinate-wise PDE metric in Appendix H.5.

**Gradient error.** We also measure errors in radial profile gradients. Let $\widehat{s}^{\,c}_{n,t}$ be the ensemble-mean estimate and $s^{*,c}_{n,t}$ the reference profile for trajectory $n$, time $t$, and channel $c$. With the forward difference $D_\rho s_j = s_{j+1} - s_j$, the reported metric is

$$\mathcal{E}_{\mathrm{grad}} = \frac{1}{3} \sum_{c=1}^{3} \mathbb{E}_{n,t} \left[ \frac{\|D_\rho \widehat{s}^{\,c}_{n,t} - D_\rho s^{*,c}_{n,t}\|_2}{\max(\|D_\rho s^{*,c}_{n,t}\|_2, 10^{-30})} \right]. \tag{69}$$

The uniform grid spacing cancels in the relative error. Both profile levels and gradients enter transport and equilibrium calculations. The gradient metric specifically measures radial smoothing and errors in local profile shape. It complements the state error without establishing the accuracy of downstream transport or stability predictions.

**Radial power spectrum.** For this benchmark, the spectrum metric compares average spectral power using the real-input fast Fourier transform (rFFT). For each state channel in physical units, we compute $E(k) = |\,\mathrm{rFFT}(s)(k)|^2/64^2$ for every trajectory, time, and posterior sample, then average the power over these dimensions. We compare the logarithms of the predicted and reference average spectra at $16$ logarithmically spaced angular wavenumbers in $[2\pi, 50]$, using linear interpolation in wavenumber. The reported $\mathcal{E}_{\mathrm{spec}}$ is the mean absolute difference at these points, averaged over the three state channels. This measures agreement of aggregate radial spectra. It does not measure the spectrum error of each posterior sample separately.

### H.5 Evaluation Metrics

The revised PDE table evaluates all four metrics on the same final-time ensemble for each of $N = 100$ test trajectories, using stored frame 199 for KS, 99 for Burgers', and 49 for NS. Let $x_{n,m,j}$ be member $m$ at state coordinate $j$, and $y_{n,j}$ the reference. KS and Burgers' use physical state units. NS uses the frozen dataset normalization already used by the checkpoints, including its test-split normalization reference. The full NS grid is scored, including solid-region target zeros. No additional target mask is applied to neural predictions. Scores from different state units or earlier time-averaged protocols are not interchangeable. All methods in the endpoint comparison receive the same observations with additive Gaussian noise. The observation-noise standard deviations are $0.046574$ for Burgers' and $0.196869$ for KS in physical units, and $0.1$ for NS in dataset-normalized units.

**Relative $L_2$ error.** With $C$ channels and ensemble mean $\bar{x}_n^c$, we compute

$$\mathcal{E}_{L_2} = \frac{1}{NC} \sum_{n=1}^{N} \sum_{c=1}^{C} \frac{\|\bar{x}_n^c - y_n^c\|_2}{\|y_n^c\|_2 + 10^{-8}}, \tag{70}$$

where each norm is over the spatial grid. Here $C = 1$ for the 1D systems and $C = 3$ for NS.

**Spectrum deviation.** The implemented metric compares the logarithmic power spectrum of each ensemble member with its reference spectrum according to

$$\mathcal{E}_{\text{spec}} = \frac{1}{NM} \sum_{n,m} \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \frac{|\log(E_{n,m}(k) + \epsilon) - \log(E_n^*(k) + \epsilon)|}{|\log(E_n^*(k) + \epsilon)| + \epsilon}, \qquad \epsilon = 10^{-12}. \tag{71}$$

For 1D fields, power is $|\,\mathrm{rFFT}(x)|^2/N_x^2$. The scorer uses a unit-length spatial coordinate for both systems and linearly interpolates power to 32 equally spaced wavenumbers in $[2\pi, 200]$ for KS and $[2\pi, 500]$ for Burgers'. The requested lower cutoff of 1 is clamped to the first nonzero Fourier wavenumber, $2\pi$. These are the scorer's coordinate conventions, not a statement that the KS simulation domain has length one. For NS, only the $u, v$ channels enter the energy spectrum. The power $\frac{1}{2}(|\widehat{u}|^2 + |\widehat{v}|^2)/(HW)^2$ is summed in radial Fourier shells under the scorer's $L = 2\pi$ convention. All nonzero shells are used. There is no additional cutoff at 85 or interpolation to 32 points in this 2D calculation. The tokamak table retains its separate aggregate radial-spectrum definition in Appendix H.4.

**Main-figure protocol.** The NS panels in the main figure show the first posterior sample for stochastic methods and the deterministic outputs for FNO and Transformer. Mean sample spectra are computed for each posterior member and then averaged over members and the 100 test trajectories, using the same state units and spectral conventions as the main table.

**Coordinate-wise miscalibration area.** For $q_\ell = \ell/19$, $\ell = 0, \ldots, 19$, let $I_{n,j}(q_\ell)$ be the closed central interval between linearly interpolated empirical quantiles $(1 - q_\ell)/2$ and $(1 + q_\ell)/2$ of the $M$ members. With $D$ state coordinates, the implemented factor-of-two convention is

$$\mathrm{MA} = \frac{2}{20D} \sum_{\ell=0}^{19} \sum_{j=1}^{D} \left| \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}\{y_{n,j} \in I_{n,j}(q_\ell)\} - q_\ell \right|. \tag{72}$$

Coverage is averaged across trajectories at each coordinate before taking the absolute error. This differs from pooling all coordinates before taking the absolute value. The metric follows central-interval calibration diagnostics (Chung et al., 2021). It does not establish calibration conditional on every observation history. Neural generative rows use $M = 16$, deterministic rows use $M = 1$, and classical ensemble sizes are reported with their configurations. A deterministic prediction is a point mass with zero-width intervals, which explains its MA near one.

The factor of two normalizes this metric to $[0, 1]$. Zero means exact empirical coverage at every evaluated coordinate and level, and one is the maximum. The intervals are nested as $q$ increases, so each coordinate's empirical coverage is nondecreasing. On our symmetric, equally spaced grid, the two absolute deviations at $q$ and $1 - q$ sum to at most one. Averaging the ten such pairs bounds the unscaled mean deviation by $1/2$. Multiplying by two gives the stated range.

Figure 4(b) plots mean empirical coverage $\bar{C}(q) = D^{-1}\sum_j C_j(q)$ against nominal coverage $q$, where $C_j(q) = N^{-1}\sum_n \mathbf{1}\{y_{n,j} \in I_{n,j}(q)\}$. The ideal curve is $\bar{C}(q) = q$. To show the coordinate-wise MA without cancellation between overcoverage and undercoverage, the same-color shaded band has bounds

$$L(q) = \frac{1}{D}\sum_j \min\{C_j(q), q\}, \qquad U(q) = \frac{1}{D}\sum_j \max\{C_j(q), q\}. \tag{73}$$

Its width is $U(q) - L(q) = D^{-1}\sum_j |C_j(q) - q|$, so the reported MA equals twice the mean shaded width at the 20 nominal levels. The band summarizes calibration deviations. It is not a confidence interval. The area between the mean curve and the diagonal can hide coordinate-wise errors, and a continuous shaded-area integral is not the discrete average used in our table.

**Standard empirical CRPS.** We average the marginal score over trajectories and coordinates using

$$\mathrm{CRPS} = \frac{1}{ND}\sum_{n,j}\left[\frac{1}{M}\sum_{m=1}^{M}|x_{n,m,j} - y_{n,j}| - \frac{1}{2M^2}\sum_{m=1}^{M}\sum_{r=1}^{M}|x_{n,m,j} - x_{n,r,j}|\right]. \tag{74}$$

For $M = 1$, this is the mean absolute error. The main table uses this standard empirical-distribution score, not the separate finite-ensemble correction with denominator $M(M-1)$. CRPS retains the state units, so its absolute scale is not comparable across differently scaled PDE settings.

# I Baseline Specifications

In this appendix, we provide the technical specifications of our baselines.

For Table 1, EnKF uses numerical PDE dynamics with an ensemble transform Kalman update, and EnSF uses an empirical-mixture score. FlowDAS and SDA use pretrained generative models. The earlier EKF based on proper orthogonal decomposition (POD), denoted POD-EKF, and the learned-dynamics EnKF implementations are confined to the historical comparisons described below. Baseline parameters are selected using training trajectories, and all configurations are fixed before test evaluation.

We evaluate our method against a comprehensive set of baselines spanning classical data assimilation, deterministic supervised learning, and score-based generative modeling. Unless otherwise stated, neural baselines are tuned to have comparable parameter counts to our model for fair comparison.

## I.1 Classical Data Assimilation (DA)

The revised fixed-sensor PDE table uses solver-EnKF with an ensemble transform Kalman analysis. The ensemble size is 64 for KS and Burgers' and 16 for NS. The NS ensemble size is fixed for computational cost, and inflation is selected among the original training candidates with 16 members. Anomaly inflation is 1.02 for KS and NS and 1 for Burgers', with zero added process noise. The NS filter propagates complete native solver states after initializing the shared prior through the declared geometry-aware lift.

**Numerical-failure fallback in NS.** A single rule applies to every trajectory. If any member's fixed-step forecast produces a nonfinite native state or exported normalized prediction, the entire partial ensemble forecast for that observation interval is discarded. All members retain their preceding state fields, time advances to the current observation, and the filter assimilates that observation and all later observations normally. A nonfinite state or output, or a linear-algebra failure in the Kalman analysis, retains that step's complete finite forecast ensemble. There is no member deletion, magnitude-based clipping, or truth-dependent replacement. The NS run uses $M = 16$ and inflation 1.02, with no new parameter search for this fallback. No fallback was triggered in its 100 test trajectories. Triggered intervals use identity dynamics rather than a successful PDE forecast. Their trajectory and step counts accompany the final evaluation. Input, configuration, input/output (I/O), and timeout errors are not treated as numerical fallbacks.

The POD and learned-dynamics Kalman descriptions below document the earlier experiments and separate historical diagnostics. They do not specify the solver-EnKF row of the revised main table.

**POD Kalman variants.** Proper Orthogonal Decomposition (POD) (Lumley, 1967; Berkooz et al., 1993) represents the physical field in a reduced basis. Kalman prediction and analysis then propagate a Gaussian belief in the reduced coordinates (Kalman, 1960; Willcox, 2006). This avoids storing a covariance over the full NS field. The preserved artifact hand-off does not specify all reduced-model orders or process/observation covariances, so it does not support a more detailed common configuration for every PDE.

**EnKF variants.** The stochastic EnKF (Evensen, 2009a) propagates an ensemble through its dynamics model and assimilates sparse observations with perturbed measurements. The historical FNO-EnKF sensitivity study in Table 6 uses a learned FNO (Li et al., 2021) transition with fixed network parameters. Its reported ensemble sizes, process and observation noise, and initialization are specified once in Appendix K.2. This FNO-based diagnostic is distinct from both the POD reduced-coordinate Kalman variants and the numerical solver-EnKF of Table 1. In particular, the numerical solver-EnKF uses zero added process noise, whereas the historical FNO diagnostic injects process noise internally. Covariance parameters inside a filter must not be identified with noise added to the shared evaluation observations.

### I.2 Supervised Learning Baselines

We compare probabilistic reconstruction with deterministic state estimates from supervised baselines. These models map the observation and action history $\tau_t$ to a single state estimate $\hat{s}_t$. We introduce two Transformer-based (Vaswani et al., 2017) baselines that share the same temporal encoder architecture but differ in their decoding heads for state reconstruction.

We employ a standard Transformer encoder to process the sequence of historical observations $\tau_t = (o_{[1:t]}, a_{[1:t-1]})$. The temporal dependencies are aggregated into the last token, which serves as a compact latent representation of the current system state. To enable parallel training, we input the entire sequence into the Transformer and compute loss with all tokens' outputs. The causal attention mask ensures that $\hat{s}_t$ depends only on the available history $\tau_t$, including the current observation.

Next, we introduce the decoders utilized by the two baselines.

**Transformer with a convolutional neural network (CNN).** In this baseline variant, we utilize a projection-based residual deconvolution decoder for spatial reconstruction. Specifically, the aggregated latent token is first linearly projected to the initial spatial dimension. Subsequently, the feature map is processed by $N$ layers of 1D transposed convolutions equipped with residual connections to generate the final state estimate. We denote this baseline as "Transformer" in the text for brevity.

**Transformer-FNO (FNO).** In this variant, we employ a Fourier Neural Operator (FNO) (Li et al., 2021) as the decoding head. The latent token conditions the FNO, which performs global convolutions in the spectral domain to reconstruct the continuous function of the current state directly. We denote this baseline as "FNO" in the text for brevity.

### I.3 Diffusion Generative Model-Based Baselines

**Score-based data assimilation (SDA).** We use the trajectory inference framework of Rozet & Louppe (2023), which learns a joint score-based prior over state sequences. The checkpoints used for the endpoint comparison use a variance-preserving stochastic differential equation (SDE) with 1000 training diffusion steps and exponential moving average (EMA) weights. The 1D checkpoints use a Markov-blanket length of 100, while the NS checkpoint uses time chunks of length 16. At inference, a predictor–corrector sampler applies the configured observation guidance through the observation operator. We use 16 samples per trajectory and corrector signal-to-noise ratio (SNR) 0.16 in all three settings. KS uses 1000 sampling steps, two corrector steps per iteration, guidance strength 1000, and a conditioning length of 100. Burgers' uses 500 sampling steps, two corrector steps, guidance strength 1, and conditioning length 100. NS uses 500 sampling steps, one corrector step, guidance strength 2, and conditioning length 16. These settings describe the evaluated checkpoints and frozen inference configurations. The endpoint is sampled conditional on each method's specified observation window, comprising the last 100 observations for each 1D system and the last 16 observations (stored frames 34–49) for NS.

**FlowDAS.** FlowDAS (Chen et al., 2025) is a generative data assimilation framework based on stochastic interpolants. It learns a deterministic drift function conditioned on historical states. Consistent with other baselines, we instantiate the FlowDAS drift model using a U-Net architecture.

During autoregressive inference, FlowDAS conditions on samples generated in the previous step and uses gradient guidance for sensor observations. The released setting supplies the ground-truth field at the first filtering step, information unavailable to the other methods in our protocol. We instead initialize its historical conditioning frames from the prior distribution estimated from training data, without using the test trajectory's ground-truth field. Thereafter we retain autoregressive conditioning on generated samples. This comparison evaluates a changed initialization protocol and should not be read as a measurement of the released setting with ground-truth initialization.

# J Comprehensive Experimental Results

## J.1 Computation Efficiency

**Inference cost.** We profile BFF's inference cost against the vanilla unconditional flow matching (FM) model (Figure 12). Across the commonly used range of 10 to 200 ODE steps, the TTT layer adds only marginal overhead on top of FM. Sampling cost grows with the number of flow-ODE evaluations. On the tokamak benchmark, the TTT weight update by itself accounts for only $5.11$ ms per step (top row of Table 5), while Euler integration with the recorded grid setting $50$ on top of it brings the total to $225.25$ ms. In the historical comparison with SDA (Table 4), BFF at $100$ ODE steps is about $20\times$ faster on KS, where SDA samples the full spatio-temporal trajectory at every physical step. Training is not as cheap as inference, since the inner-loop second-order derivative does add nontrivial cost, but the overhead remains moderate, with the additional training-step cost staying within $\sim 30\%$ of vanilla flow matching in a large-batch regime (effective batch size $50$, Figure 11).

**Integration resolution and mean error.** Reducing the integration resolution lowers the sampling cost without necessarily increasing posterior-mean error. In the historical KS and NS comparisons (Table 4), reducing sampling from $100$ to $2$ steps lowers latency from $415$ to $11$ ms and from $644$ to $23$ ms, respectively. The corresponding relative $L_2$ errors also decrease slightly, from $0.9061$ to $0.8819$ on KS and from $0.165$ to $0.162$ on NS. On tokamak, Euler with five grid nodes matches the reported mean error of $50$ nodes while reducing latency from $225.25$ to $18.66$ ms (Table 5). They do not establish a general improvement from coarser integration.

**ODE solver and calibration.** On tokamak, reducing Euler from $50$ to five grid nodes preserves mean accuracy but increases MA from $0.0374$ to $0.2529$. At two nodes, MA rises further to $0.6219$ and mean error also increases. Thus fewer Euler nodes exchange calibration quality for faster sampling, even over a range where mean error is unchanged. Solver choice improves this tradeoff. Five midpoint nodes retain the same mean error as $50$ Euler nodes while reducing latency to $36.92$ ms and MA to $0.0225$. Reducing midpoint to two nodes further accelerates sampling and lowers spectrum error, but increases both mean error and MA. We use five midpoint nodes for the main tokamak results. Each update takes $36.92$ ms, but the dataset provides an observation every $5$ ms. This configuration is therefore too slow to process every observation as it arrives.

## J.2 Ablation of the TTT and GRU Belief Updates

Vectors in latent space can also encode belief states without the need of a large number of particles. We ablate the TTT block by replacing it with a flow matching model conditioned on a GRU hidden state ("hidden-state-based"), keeping the backbone, training data, and shared hyperparameters identical. As shown in Figure 13, the TTT block improves performance in this 2D-flow comparison. The result is architecture- and task-specific. It does not prove a general representational limit for recurrent hidden states.

## J.3 Multimodal Posterior Visualization

Figure 3 in the main text shows posterior samples for Burgers' and KS with four spatial sensors and for Lorenz-63 with only $Z$ observed. Dataset and observation details are in Appendix H.

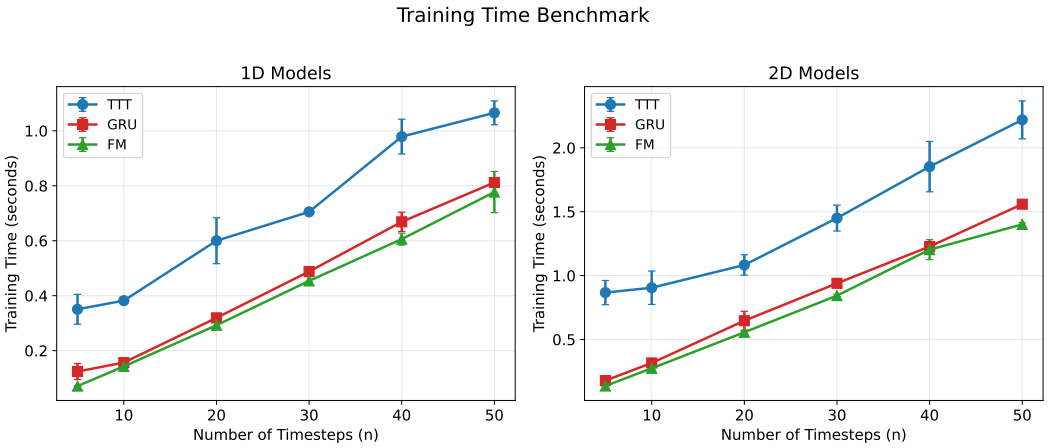



Figure 11: **Training wall-clock time comparison.** BFF, the gated recurrent unit (GRU) variant, and the vanilla flow matching (FM) model across different numbers of unrolled timesteps.

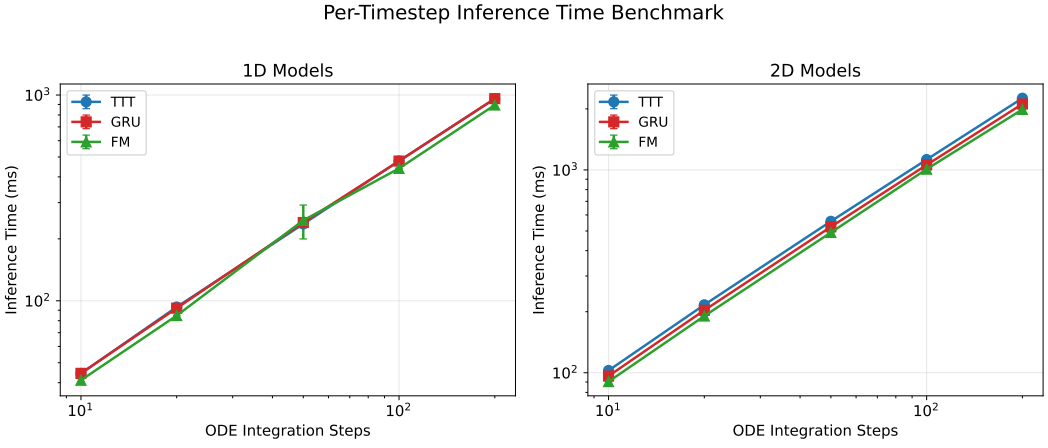



Figure 12: **Inference time comparison.** BFF, the GRU variant, and the vanilla FM model across different ODE integration steps.

Table 4: **Neural inference time with the main-table configurations.** Checkpoints, observations, preprocessing, and sampler settings follow Table 1, except for the additional two-step BFF variant. Errors use the main-table evaluation protocol and posterior sample counts (BFF: 100 for KS, 16 for NS), with the two-step BFF variant evaluated on the same 100 test trajectories per setting; timings generate *one* sample on a single NVIDIA H800 and average three trajectories after one warmup, with CUDA synchronization. Model and input loading are excluded. BFF includes the current observation update and sampling from its cached history state; FNO and SDA use their causal history and final observation window, respectively. BFF step labels count Euler grid nodes (one fewer flow evaluation).

| Method | KS Equation | | NS Equation | |
|---|---|---|---|---|
| | $\mathcal{E}_{L_2}\downarrow$ | Time (s) | $\mathcal{E}_{L_2}\downarrow$ | Time (s) |
| FNO | 0.9756 | 0.007 | 0.3281 | 0.016 |
| SDA | 0.9345 | 53.071 | 0.1608 | 46.762 |
| BFF (100 steps) | 0.8710 | 0.581 | 0.2250 | 0.585 |
| BFF (2 steps) | 0.8710 | 0.020 | 0.2431 | 0.022 |

Table 5: **Integration resolution and solver choice on the tokamak benchmark.** The grid setting is the stored `ode_steps` argument. The evaluator constructs this many equally spaced nodes, so the five-node midpoint configuration integrates four intervals. The setting does not count vector-field evaluations. The first row reports the cost of the TTT weight update alone (no ODE integration). Per-step inference latency is measured on a single NVIDIA H800 graphics processing unit (GPU). **Bold** denotes the best in each column.

| Solver | Grid setting | ms/step ↓ | $\mathcal{E}_{L_2}\downarrow$ | $\mathcal{E}_{\mathrm{grad}}\downarrow$ | $\mathcal{E}_{\mathrm{spec}}\downarrow$ | MA ↓ |
|---|---|---|---|---|---|---|
| TTT W update only (no ODE) | | 5.11 | — | — | — | — |
| Euler | 50 | 225.25 | **0.0193** | **0.0471** | 0.0102 | 0.0374 |
| Euler | 10 | 49.88 | **0.0193** | 0.0476 | 0.0108 | 0.1233 |
| Euler | 5 | 18.66 | **0.0193** | 0.0490 | 0.0115 | 0.2529 |
| Euler | 2 | 5.30 | 0.0197 | 0.0717 | 0.0150 | 0.6219 |
| Midpoint | 5 | 36.92 | **0.0193** | 0.0486 | 0.0101 | **0.0225** |
| Midpoint | 2 | 9.61 | 0.0198 | 0.0714 | **0.0095** | 0.1157 |

On the tokamak benchmark, pooled temperature residuals show non-Gaussian marginal shapes and correlations that vary with radius (Appendix J.7). Because this diagnostic combines different observation histories, it does not establish non-Gaussianity of each conditional posterior.

### J.4 Full Results on Lorenz-63 System

The Lorenz-63 experiments test recursive filtering beyond the training horizon. Figure 15 follows a posterior with two symmetric modes to $T_{\text{test}} = 1000$ ($10\times$ the training horizon), while Figure 14 compares the generated beliefs with a $10^4$-particle bootstrap reference to $T = 4000$ ($40\times$ training). These tests evaluate the learned update under repeated assimilation of fresh observations. A recursive Markov form alone does not ensure accurate one-step updates or prevent their errors from accumulating.

**Results.** Figure 14 reports the Wasserstein-2 distance between BFF's posterior and a bootstrap particle filter (BPF) with $10^4$ particles, averaged over 10 test trajectories of length $T = 4000$. The plotted mean $W_2$ distance falls during initial assimilation and then fluctuates rather than increasing steadily. At steps 100, 2000, and 3000, the generated samples remain split between the true state and its sign-symmetric counterpart. These snapshots show that BFF retains the observation-induced ambiguity far beyond the training horizon. Figure 15 visualizes the evolving posterior over a $T_{\text{test}} = 1000$ trajectory. BFF captures two symmetric modes, with one mode tracking the ground truth. A separate 99-point diagnostic through step 990 finds a weak positive correlation between $W_2$ and time (Appendix K). These observations do not establish absence of error accumulation or exact posterior recovery.

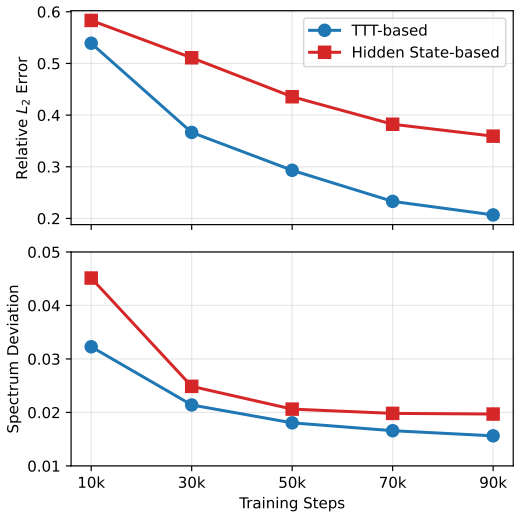



Figure 13: **Detailed training evolution of the ablation study on 2D Navier-Stokes.** The TTT variant has lower tracking and spectrum errors than the GRU variant in this comparison.

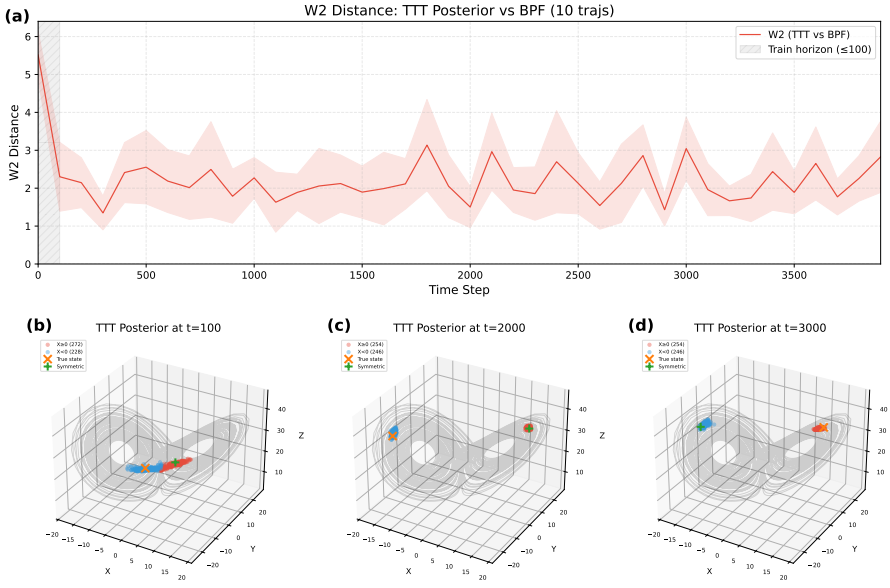



Figure 14: **Long-horizon Lorenz-63 diagnostic.** Wasserstein-2 distance between a bootstrap particle filter (BPF) reference with $10^4$ particles and the posterior predicted by BFF through $T = 4000$, $40\times$ the training horizon, averaged over 10 trajectories. Insets show generated samples alongside the true state and its sign-symmetric counterpart at selected steps. The particle filter is an empirical reference rather than an exact posterior.

### J.5 Trajectory Visualization on the 1D KS Benchmark

Figure 16 compares the posterior mean with the ground-truth KS field over a representative test rollout. The reconstruction captures visible traveling structures and the cellular spatial pattern, while retaining appreciable discrepancies in their positions and amplitudes. The recognizable spatial pattern thus coexists with substantial pointwise error in this sparse-observation task. The figure does not separate the contributions of position and amplitude errors.

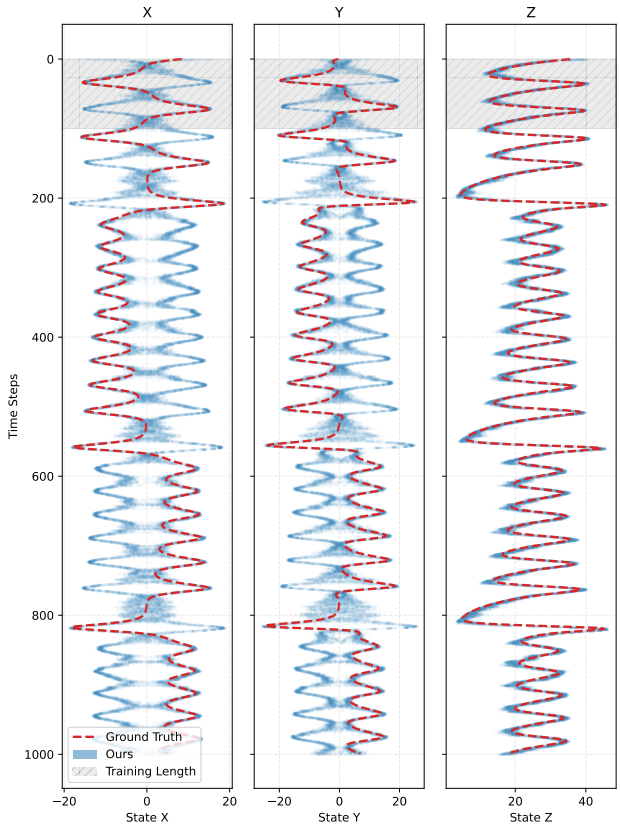



Figure 15: **Trajectory visualization on Lorenz-63.** 100 samples per step (darker = higher density). BFF captures the two symmetric modes, one consistently tracking ground truth, with uncertainty inflating near lobe switches.

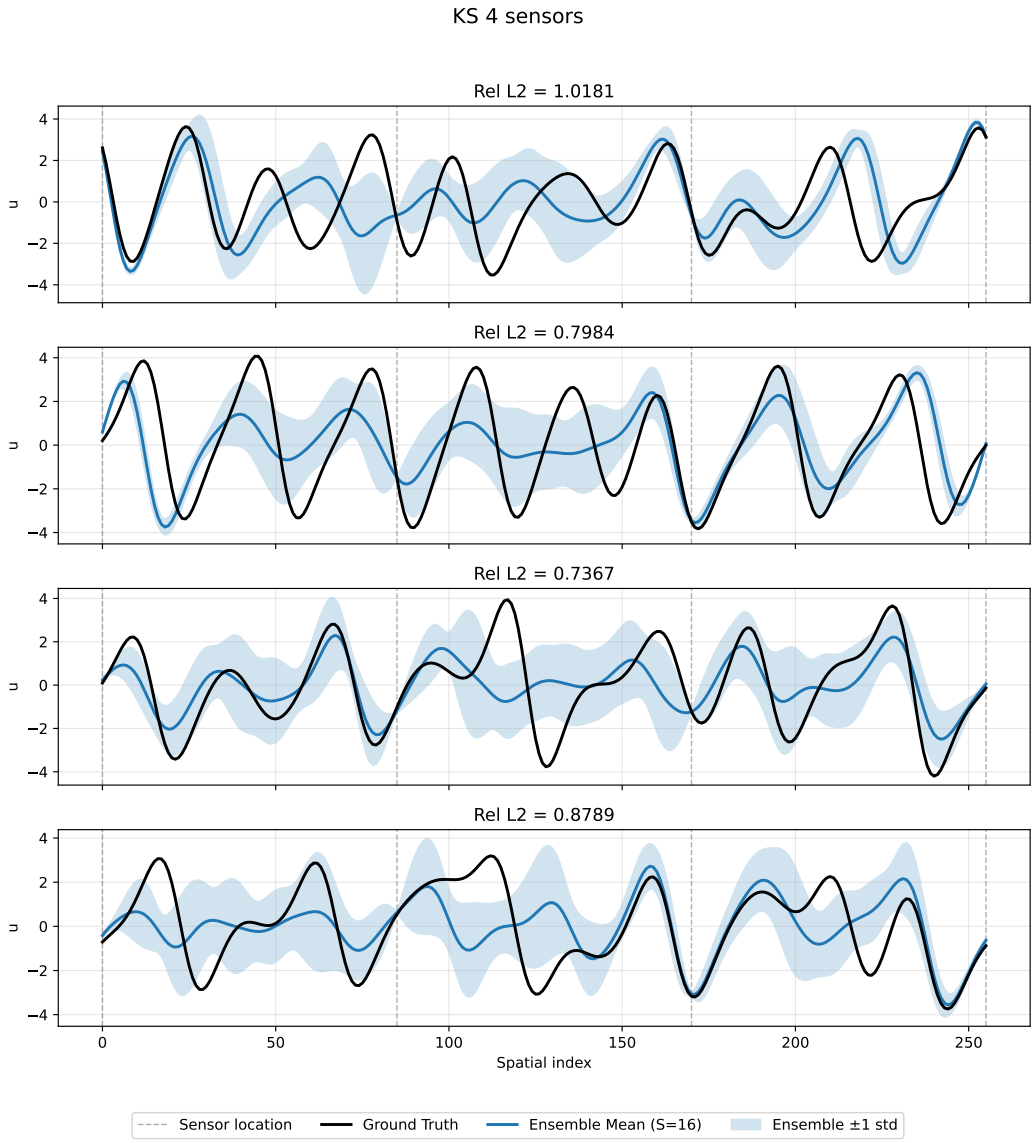



Figure 16: **Trajectory visualization on the 1D KS benchmark.** Space-time fields of the ground truth and the BFF posterior mean over a representative test rollout. The mean recovers visible traveling structures and cellular patterns, with discrepancies in position and amplitude.

### J.6 Full Results on 2D NS System

The uniform-sensor Navier–Stokes comparison appears in Table 1. The moving-sensor results appear in Table 2.

**Moving-sensor experiment.** A single sensor follows a fixed zig-zag path through the flow domain (Figure 6). The filter reconstructs the $128 \times 128$ field from measurements at this one spatial location at each step. The same prescribed path is used for all test trajectories. Table 2 retains the original moving-sensor comparison. This experiment is separate from the common endpoint evaluation of the three fixed-sensor benchmarks.

We also inspect the spatial structure of the reconstructed fields. Figure 18 shows interacting wakes behind multiple cylinders. In this example, the BFF reconstruction retains sharper wake structures

than the FNO prediction. This visual comparison complements the spectrum metric. It does not by itself test conservation laws or establish posterior correctness.

Under the single moving sensor setting, Figure 19 compares BFF with the Transformer baseline on a representative trajectory. The BFF sample more closely matches the obstacle locations and surrounding flow in this example. Figure 20 tracks binomial entropy and ensemble standard deviation for the predicted cylinder maps and wake regions. Their decline as observations arrive is consistent with reduced spatial ambiguity. The sensor follows a prescribed path, so this experiment evaluates sequential assimilation rather than learned active sensor placement.

Figure 21 compares the spatial reconstruction error with the standard deviation of vorticity across generated samples. The displayed maps show qualitative correspondence in high-error and high-uncertainty wake regions. This is a useful spatial diagnostic, but no correlation coefficient or joint-posterior reference is provided by the figure. Marginal calibration is assessed separately by the reported probabilistic metrics.

Figure 22 reports the state estimation error over filtering steps. The error decreases during the initial assimilation period and reaches an approximately steady level after about 10 steps. This supports sustained tracking over the plotted horizon. The Markov property does not imply that distant observations are irrelevant, and this plateau does not establish a general absence of error accumulation. The additional horizon tests are reported separately in Appendix K.

### J.7 Temperature Residuals on the Tokamak Benchmark

Figure 17 examines electron-temperature residuals at four radial coordinates, pooling $20{,}000$ samples across trajectories, prediction horizons, and time steps. The pooled marginals differ from Gaussian fits, and residual correlations vary across the radial pairs. Pooling can produce non-Gaussian distributions even when each conditional posterior is Gaussian. A Gaussian joint distribution can also represent correlations. This figure therefore describes the aggregate residuals without establishing the shape or accuracy of each conditional posterior.

## K Additional Evaluation and Scope Checks

The following experiments were conducted after the initial evaluation to test recursive-filter baselines, posterior scores, variability, update capacity, and sensitivity to the observation protocol. The directly comparable EnSF results appear in the main benchmark table, while results from separate diagnostic protocols remain in this appendix.

### K.1 A Recursive Ensemble Score Filter Baseline

EnSF (Bao et al., 2024) is a recursive, score-based filter. The fixed-sensor endpoint comparisons in Table 1 use numerical dynamics for the EnSF forecast and an empirical-mixture score, with ensemble sizes $64$, $16$, and $16$ for KS, Burgers', and NS, respectively. BFF has lower errors than EnSF on the reported metrics except KS MA, where EnSF is lower ($0.2017$ versus $0.2094$). These point estimates do not establish a seed-level significance claim.

### K.2 Historical Diagnostic Protocols

The following sensitivity results document earlier evaluation protocols and baseline implementations. They are separate from the common endpoint comparison in Table 1. Its numerical-dynamics filters and selected settings are described in Appendix I.

**Earlier baseline search and EnKF sensitivity.** The earlier performance comparisons used each method's selected configuration. The historical efficiency table reports reconstruction error and latency for the recorded implementations, including BFF configurations with $100$ and $2$ sampling steps. Under that earlier protocol, the generative-baseline search included guidance strengths $\{0.05, 0.1, 0.5, 1, 5, 10, 50, 500\}$, gradient clipping thresholds $\{0.1, 1, 10, 1000\}$, ODE or diffusion solver steps $\{10, 50, 100, 500, 1000\}$, and observation-covariance regularization values $\{0.1, 1\}$. This records the searched candidates, not a claim that every candidate was evaluated for every baseline

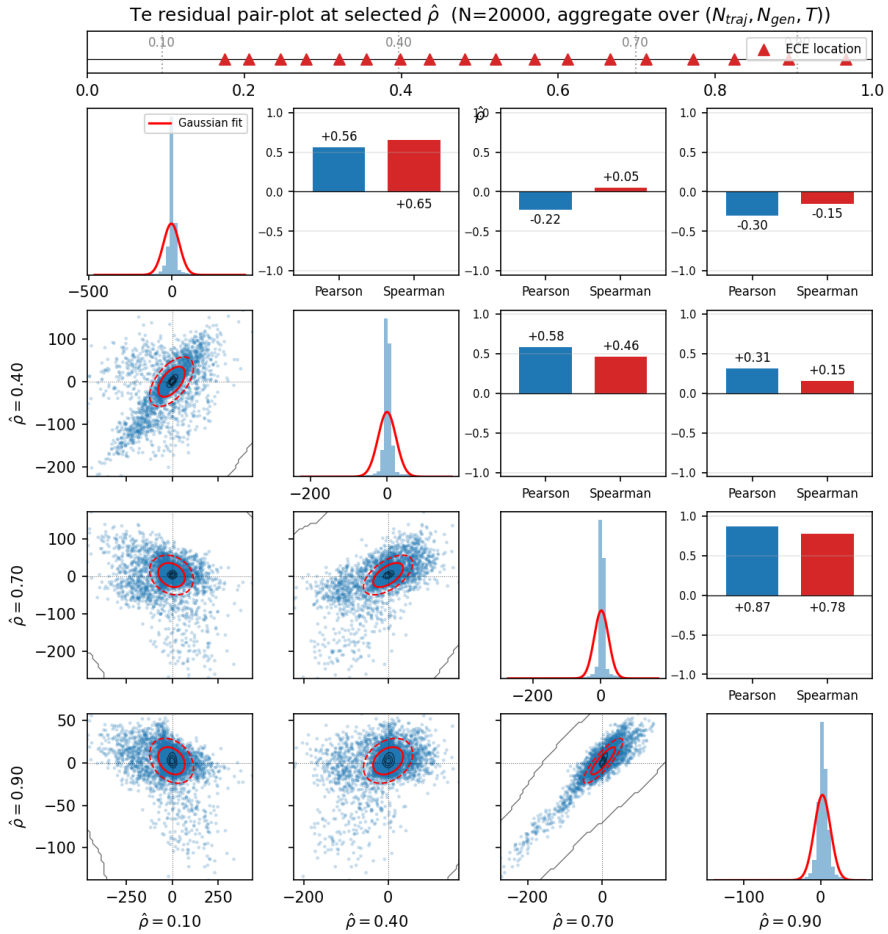



Figure 17: **Pooled temperature residuals on the tokamak benchmark.** Pair-plot of $T_e - \widehat{T}_e$ at $\hat{\rho} \in \{0.10, 0.40, 0.70, 0.90\}$, using 20,000 samples across trajectories, prediction horizons, and time steps. Red triangles in the top strip mark ECE locations. The diagonal shows marginal histograms and Gaussian fits. The lower triangle shows residual pairs, and the upper triangle reports Pearson and Spearman correlations. The pooled marginals have sharper peaks and heavier tails than the fits, with stronger correlations between the two outer radial positions.

and task. The current main-table settings, including FlowDAS initialization, are specified separately in Appendix I. The historical search ranges below are not a record of the final selected endpoint configurations.

In the earlier FNO-EnKF implementation, each physical step propagates every ensemble member through the trained FNO and then applies one stochastic EnKF analysis with perturbed observations. FNO parameters remain fixed. BFF instead updates its compact per-layer TTT weights once and does not evaluate a dynamics model during that online update. The rebuttal reports EnKF ensemble sizes 88, 40, and 40 for 2D NS, Burgers', and KS, respectively. Their process-noise standard deviations are 0.0039, 0.2, and $10^{-5}$, and their observation-noise standard deviations are 0.0072, $10^{-4}$, and $10^{-4}$, respectively, with no localization. A separate 2D NS one-at-a-time diagnostic varied ensemble size on a fixed $N = 30$ subset, with final-step fair CRPS (Table 6). The uniform-sensor result improves with ensemble size, while moving-sensor relative $L_2$ changes little. The chosen 88 is a practical accuracy-cost choice rather than a universally optimal ensemble size. The diagnostic's noise scales are in normalized state space, and its stored 2D NS observations have no added noise. The covariance is injected inside EnKF. These details describe this diagnostic protocol and should not be conflated with every dataset-generation setting in the main experiments.

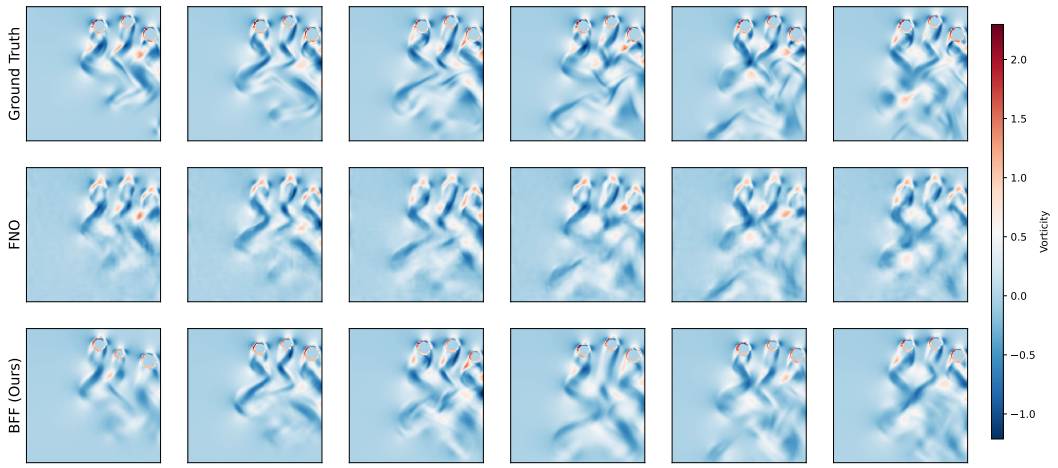



Figure 18: Qualitative comparison of 2D Navier-Stokes vorticity fields between the ground truth, FNO, and BFF.

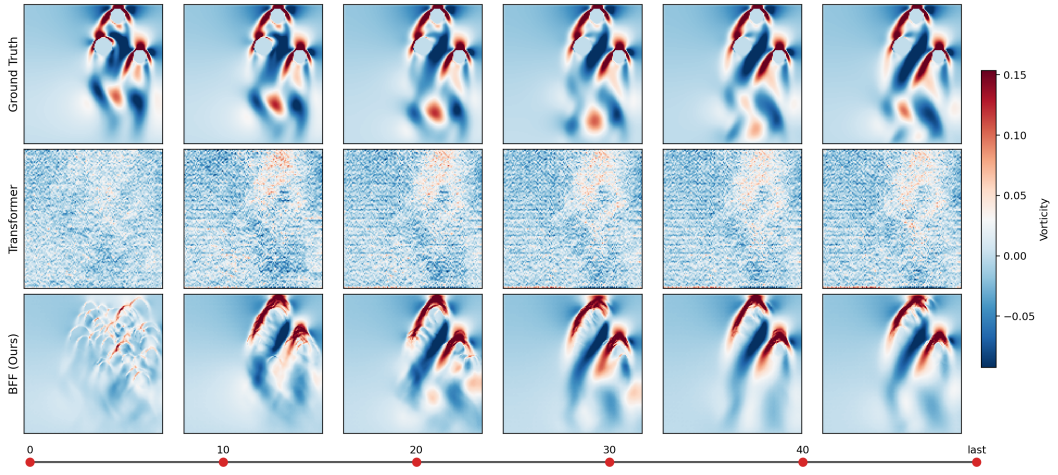



Figure 19: Qualitative comparison of single-sample predictions along a moving-sensor trajectory. The final $L_2$ error is $0.330$ for BFF and $0.769$ for Transformer.

Table 6: FNO-EnKF ensemble-size sensitivity on the separate 2D NS $N = 30$ diagnostic subset. All metrics are lower-is-better.

| Ensemble | Moving relative $L_2$ | Moving CRPS | Uniform relative $L_2$ | Uniform CRPS |
|---|---|---|---|---|
| 20 | 0.7815 | 0.2783 | 1.0503 | 0.3618 |
| 40 | 0.7781 | 0.2767 | 1.0240 | 0.3531 |
| 88 | 0.7739 | 0.2718 | 0.8968 | 0.3095 |
| 128 | 0.7726 | 0.2720 | 0.8477 | 0.2934 |

### K.3 KS Robustness Across Training Seeds

Table 7 evaluates five independently trained BFF models, with training seeds $0$, $1$, $2$, $3$, and $42$. Each uses the checkpoint at training step $15{,}000$, $16$ posterior samples per trajectory, and $100$ Euler grid nodes ($99$ integration intervals). We hold the $100$ test trajectories, noisy observation realizations, normalization reference, final evaluated frame, and inference random seed fixed across the five models. All four metrics are computed from the same final-time ensemble at stored frame $199$. They are not

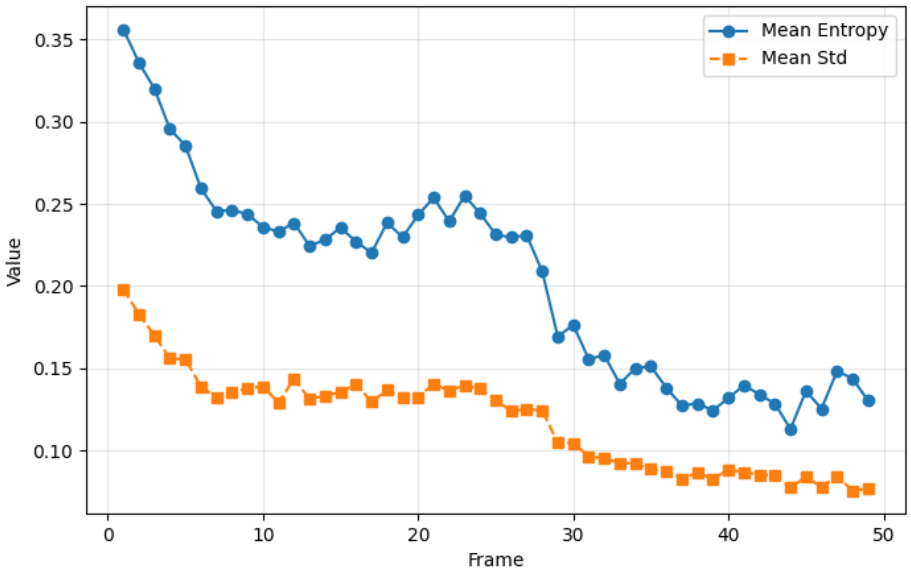



Figure 20: Evolution of spatial uncertainty under the single moving sensor setting. The uncertainty is quantified by tracking the binomial entropy and the ensemble standard deviation of the predicted probability maps for the cylinders and their associated wake regions.

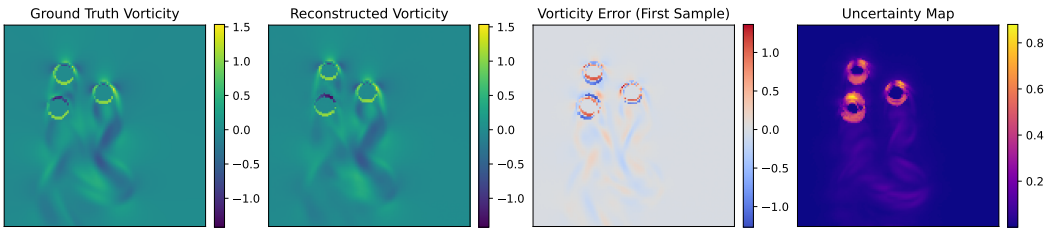



Figure 21: Qualitative comparison between the spatial error field and the predicted uncertainty map (standard deviation of sample vorticities) for the 2D Navier-Stokes benchmark.

averaged over filtering times. MA uses coordinate-wise coverage with linear empirical quantiles and the factor-of-two convention, and CRPS is the standard empirical marginal score.

Across these five training seeds, the standard deviations of relative $L_2$ error and CRPS are approximately $1\%$ and $2\%$ of their means, compared with $10\%$ for MA. Calibration is therefore more sensitive to the training seed than mean reconstruction accuracy under this fixed evaluation protocol. The sample standard deviations quantify variation across the trained models. They do not measure posterior uncertainty, variation across new test sets, or the significance of a difference between methods. This experiment tests BFF's training-seed sensitivity. It is not a five-training-seed comparison of all baselines. Repeated inference seeds for a numerical filter would measure a different source of variability.

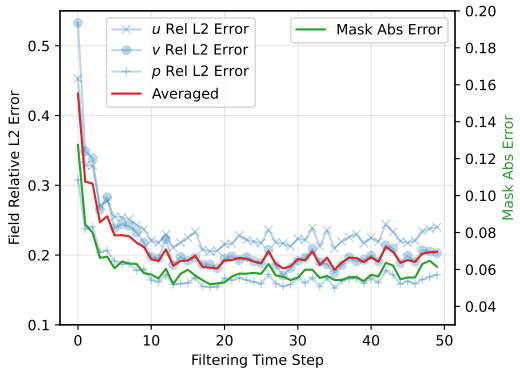



Figure 22: Temporal evolution of the state estimation error over filtering time steps for the 2D Navier-Stokes system.

Table 7: BFF on KS across five training seeds under a fixed endpoint protocol. All metrics are lower-is-better. The last two rows give the mean and sample standard deviation (SD, $n-1$ denominator) across trained models.

| Training seed | Relative $L_2$ | Spectrum | MA | CRPS |
|---|---|---|---|---|
| 0 | 0.871031 | 0.554926 | 0.209383 | 0.958790 |
| 1 | 0.861431 | 0.559239 | 0.204981 | 0.944211 |
| 2 | 0.870688 | 0.582251 | 0.172544 | 0.952173 |
| 3 | 0.884659 | 0.541362 | 0.226340 | 0.984708 |
| 42 | 0.874826 | 0.579011 | 0.198903 | 0.962731 |
| Mean | 0.872527 | 0.563358 | 0.202430 | 0.960523 |
| Sample SD | 0.008385 | 0.017131 | 0.019572 | 0.015237 |

### K.4 Update Architecture and Number of Inner Steps

Theorem 3.1 concerns the outer loss and also applies to a recursively updated GRU belief encoder. It does not rank the architectures. In a separate 2D NS moving-sensor comparison, a two-layer GRU modulated the same frozen DiT backbone and used the same training data and other configurations. The TTT model had 64.7 million parameters and relative $L_2$/spectrum/MA of $0.3771/0.0286/0.1799$. The larger GRU variant with 78.5 million parameters obtained $0.6090/0.0633/0.3901$. This is empirical support for the chosen update on this task, not a general impossibility result for fixed-dimensional recurrent states.

The practical surrogate uses a fixed learned probe $x$ to read each online matrix $W$ through $Wx$. Since the loss depends on $W$ through this contraction, its gradient with respect to $W$ has rank at most one per probe, and the update remains in the corresponding fixed right-factor subspace over time. Multiple probes expand that subspace but do not make the idealized full-weight construction of Proposition 4 applicable automatically. Table 8 shows that increasing the probe count did not improve all metrics monotonically. The participation rank of the normalized learned probe Gram matrix remained below 2 even at $k = 8$. The finite-capacity gap remains unquantified.

Table 8: Probe-count sweep on 2D NS with uniform sensors under a separate diagnostic protocol. The $k = 1$ column is the baseline of this sweep, not a repetition of the endpoint row in Table 1. Lower is better except for effective rank.

| Metric | $k = 1$ | $k = 2$ | $k = 4$ | $k = 8$ |
|---|---|---|---|---|
| Relative $L_2$ | 0.1637 | 0.1691 | 0.1642 | 0.1778 |
| Spectrum | 0.01629 | 0.01656 | 0.01625 | 0.01662 |
| MA | 0.03038 | 0.02953 | 0.02215 | 0.02625 |
| CRPS | 0.02569 | 0.02587 | 0.02580 | 0.02770 |
| Effective probe rank | 1.00 | 1.35 | 1.65 | 1.86 |

We also changed the number of gradient steps at inference without retraining the one-step checkpoint, and varied the physical-time stride between observations. Here an inner step means a TTT weight update, not a flow-ODE sampling step; stride counts the saved-frame intervals between successive observations. The one-update, stride-one result is the reference within this diagnostic, not the current main-table endpoint evaluation. Table 9 reports the relative $L_2$ and spectrum results from this diagnostic protocol. Additional inner steps worsen both metrics in the moving-sensor NS case at all tested strides. On KS, they worsen relative $L_2$ at stride $1$ but improve it at stride $5$, while spectrum error changes less. The effect therefore depends on the task and observation interval. Because the checkpoint was trained for one step, these numbers should not be interpreted as an optimized comparison of separately trained multi-step models.

Table 9: Relative $L_2$ and spectrum errors when applying different numbers of inner updates to a one-step-trained checkpoint under a separate stride diagnostic protocol. The last column uses four updates for NS and five for KS.

| System | Metric | Stride | 1 step | 2 steps | 4 or 5 steps |
|---|---|---|---|---|---|
| NS, moving | $\mathcal{E}_{L_2}$ | 1 | 0.6774 | 1.7519 | 7.1692 |
| | | 2 | 0.6813 | 1.5734 | 1.7346 |
| | | 4 | 0.7305 | 0.8887 | 1.6771 |
| | | 8 | 0.6491 | 0.6609 | 0.8483 |
| | $\mathcal{E}_{\text{spec}}$ | 1 | 0.0896 | 1.0607 | 1.0658 |
| | | 2 | 0.0522 | 0.8792 | 1.0518 |
| | | 4 | 0.0687 | 0.2055 | 0.9982 |
| | | 8 | 0.0594 | 0.0617 | 0.2186 |
| KS | $\mathcal{E}_{L_2}$ | 1 | 0.1544 | 0.2960 | 0.2352 |
| | | 5 | 0.8632 | 0.5451 | 0.3704 |
| | $\mathcal{E}_{\text{spec}}$ | 1 | 0.2800 | 0.2581 | 0.2513 |
| | | 5 | 0.2785 | 0.2827 | 0.2730 |

### K.5 Observation Shift, Horizon, and Training Cost

The deployed update does not receive an arbitrary test-time likelihood model. Dependence on the training likelihood is amortized into the learned parameters. In the moving-sensor experiment the observed coordinates are supplied to the update, so it conditions on measurement location, while the observation-noise model is fixed during training. A stress test changed test noise on the same $100$ trajectories while keeping the state trajectories, observation operator, sensor path, and inference budget fixed. The checkpoint trained at $\sigma_{\text{train}} = 0$ had mean relative $L_2$ errors of $0.426$, $0.434$, and $0.452$ at $\sigma_{\text{test}} = 0$, $0.025$, and $0.05$, respectively. This shows limited mean-error robustness and says nothing about whether predictive uncertainty adjusts correctly to an unseen noise model. A separate evaluation trained only on zigzag sensor paths and tested on unseen paths (Table 10). Accuracy degraded relative to the seen path. This tests observation-geometry shift, not arbitrary likelihood-shift correction.

Table 10: Moving-sensor path shift with fixed state trajectories, measurement rule, noise model, and inference budget. Only zigzag paths appeared in training. Lower is better.

| Sensor path | Seen in training | Relative $L_2$ |
|---|---|---|
| Zigzag | yes | 0.463 |
| Lissajous | no | 0.620 |
| Spiral | no | 0.578 |
| Curiosity-driven | no | 0.703 |
| Random | no | 0.617 |

In a separate long-horizon diagnostic on eight independently generated trajectories, the same checkpoint gave endpoint relative $L_2$/MA/spectrum of $0.660/0.463/0.070$, $0.640/0.456/0.112$, and $0.681/0.444/0.097$ at horizons $50$, $100$, and $200$, respectively. The metrics do not degrade monotonically over this range. No contraction or stability guarantee follows beyond it. On Lorenz-63, the empirical $W_2$ distance to a $10^4$-particle bootstrap reference, evaluated at $99$ points from step $10$ to $990$, had temporal mean $2.3168$, minimum $0.9657$, maximum $5.0164$, and Pearson/Spearman correlations with time of $0.223/0.220$. The weak positive association is compatible with some long-horizon degradation even though the measured distance does not grow monotonically. The particle approximation is a reference, not an exact posterior oracle.

Training horizons of $25$, $50$, and $100$ cost $1.18$, $2.38$, and $4.91$ seconds per iteration, with $16.0$, $31.2$, and $61.7$ gigabytes (GB) peak memory. No not-a-number (NaN) or infinity (Inf) events occurred. At resolutions $64^2$ and $128^2$, training iterations took $2.34$ and $2.40$ seconds, peak memory was $9.8$ and $31.2$ GB, and online updates took $0.92$ and $1.19$ seconds, respectively. These measurements do not establish resolution-independent accuracy or stability beyond the tested horizons.