Rethinking Forward Processes for Score-Based Nonlinear Data Assimilation in High Dimensions

Eunbi Yoon[1] **Donghan Kim**[1,†] **Dae Wook Kim**[1,2,3,†]

[1]KAIST, [2]KIHAC, [3]KAIST-FORMOSA Bio R&D Center
{eunbiyoon6286, kimdonghan, daewook}@kaist.ac.kr

Abstract

Data assimilation is the process of estimating the state of a dynamical system over time by combining model predictions with measurements. This task becomes challenging when the system is nonlinear and high-dimensional. To address this, score-based Bayesian filters have recently emerged. However, their performance can degrade under spatially sparse or nonlinear measurements. Such degradation stems from heuristic approximations of the likelihood score, whose errors can accumulate over time. This limitation arises because the methods simply adopt a classical forward process that transforms a data distribution toward a Gaussian distribution, which is independent of the measurement equation. Here, we propose a forward process tailored for filtering that transforms states to the measurement space, enabling a theoretically grounded formulation of the likelihood score. Based on this, we develop the Measurement-Aware Score-Based Filter (MASF). We evaluate MASF on Lorenz96 and Kolmogorov flow under diverse linear and nonlinear measurement operators, with the latter benchmark reaching state dimensions on the order of 10^5. MASF shows improved performance over existing ensemble-type Kalman filters and score-based filters. MASF also achieves up to a $20.7\times$ online wall-clock speedup compared with baselines when using amortized pretraining. Our implementation is available at `https://github.com/tcnllab-oss/masf`.

1 Introduction

Data assimilation estimates the state of a dynamical system over time by combining model predictions with measurements [1, 2]. It arises in a broad range of domains where time-evolving dynamics must be inferred from incomplete measurements, including geophysical forecasting and biological processes [3–5]. This problem is typically solved using Bayesian filtering, which alternates between a time-update step that propagates the current state under the state equation and a measurement-update step that corrects the prediction using measurements from the measurement equation [6]. However, carrying out these updates is rarely feasible in high-dimensional nonlinear systems because they involve integrals that typically lack closed-form expressions [6].

To circumvent this intractability, various classes of Bayesian filters have been proposed. Building on the Kalman filter [7], ensemble-based Gaussian filters, such as the ensemble Kalman filter (EnKF) and the local ensemble transform Kalman filter (LETKF) [8–11], approximate the posterior under Gaussian assumptions by propagating ensemble-based moment estimates. Although computationally efficient, their accuracy can degrade when the posterior is strongly non-Gaussian or when the state and measurement equations are highly nonlinear [10, 12]. In contrast, particle filters, including the bootstrap filter [13] and the auxiliary particle filter [14], approximate the posterior with weighted samples and can represent non-Gaussian distributions more flexibly, but suffer from weight degeneracy in high-dimensional settings [15, 16]. Transport-based filters, such as the stochastic map filter [12] and the optimal transport particle filter [17, 18], can alleviate these high-dimensional difficulties

[†]Corresponding authors.

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

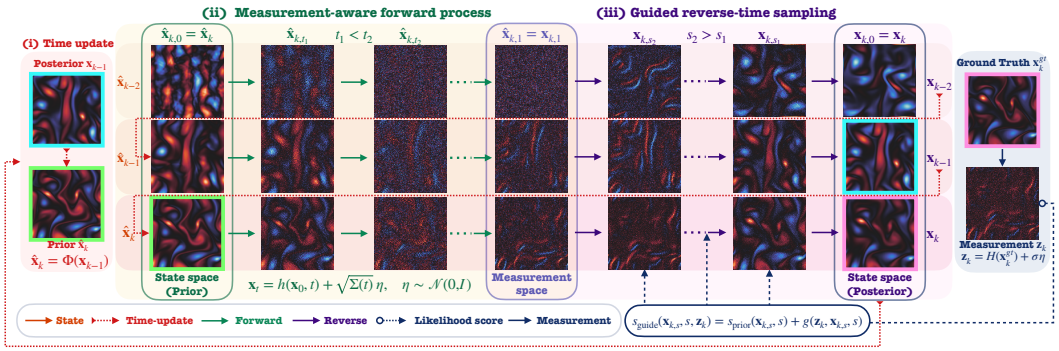


Figure 1: **Overview of MASF.** The measurement-aware forward process interpolates states toward the measurement space and the guided reverse-time sampling performs the measurement update.

by transforming prior samples to approximate non-Gaussian posterior distributions. However, their scalability often relies on problem-specific structural assumptions [12, 18].

Recently, score-based generative modeling has emerged as another approach to data assimilation [19–22], drawing on its ability to sample from high-dimensional distributions by learning score functions, i.e., gradients of log-densities [23–27]. Specifically, Score-based Data Assimilation [19] performs trajectory inference, while Score-Based Filter (SF) and Score-based Sequential Langevin Sampling (SSLS) perform Bayesian filtering. These methods generate posterior samples by combining a prior score learned on perturbed states with a likelihood score and have demonstrated scalability to high-dimensional systems.

Although these results are promising, there is still room for improvement. For example, SF uses a forward process to transform the state distribution into a Gaussian distribution [20], making the likelihood score at the resulting perturbed states generally intractable. To estimate this score, SF adopts a heuristic single point-mass approximation with a damping function, which can lead to degraded performance. SSLS avoids this likelihood-score approximation by using Langevin dynamics to sample from the posterior distribution, combining the original likelihood score with a learned prior score. However, accurate posterior sampling requires this learned score to generalize from the prediction distribution to the distributions visited during Langevin sampling [22].

Here, we propose the Measurement-Aware Score-Based Filter (MASF), which incorporates the measurement equation directly into the forward process. From this process, we derive an exact likelihood score at perturbed states for linear measurements and a theoretically grounded approximation for nonlinear measurements. Furthermore, MASF can achieve lower computational cost than ensemble-based Kalman filters in high-dimensional settings through amortized pretraining and a lightweight model. This provides a potential solution to the computational burden of retraining the score model at each measurement time, which has long been a major obstacle to the scalability of score-based filtering.

Main contributions

- We introduce a measurement-aware forward process that interpolates between the state and measurement spaces according to the measurement equation, allowing the learned prior score to reflect the state-measurement relationship (Fig. 1).
- We derive an exact likelihood score for linear measurements and a theoretically grounded approximation for nonlinear measurements. The derivation uses Markovian projection and, for linear measurements, a linear stochastic differential equation (SDE) with the same marginal distributions as the forward process.
- We demonstrate on Lorenz96 and Kolmogorov flow that MASF improves accuracy. On Kolmogorov flow, MASF also improves wall-clock efficiency and scales to the order of $\mathcal{O}(10^5)$ dimensions under nonlinear and dimensionally mismatched measurements.

2 Background

2.1 Bayesian filtering

Let τ denote the physical time variable associated with the underlying dynamical system. Let $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ and $g : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^{d\times d}$ denote the drift and diffusion terms, respectively. We consider a continuous-time state process $X_\tau \in \mathbb{R}^d$ governed by the state equation, written as a stochastic differential equation (SDE):

$$dX_\tau = f(X_\tau, \tau)\, d\tau + g(X_\tau, \tau)\, dB_\tau, \tag{1}$$

where B_τ denotes a d-dimensional Brownian motion. Let $\{\tau_k\}_{k=1}^K$ denote the discrete measurement times, with $X_k := X_{\tau_k}$. At each measurement time τ_k, we observe a measurement $Z_k \in \mathbb{R}^m$ through the measurement equation

$$Z_k = H(X_k) + \varepsilon_k, \qquad \varepsilon_k \sim \mathcal{N}(0, R), \tag{2}$$

where $H : \mathbb{R}^d \to \mathbb{R}^m$ denotes the measurement operator and $R \in \mathbb{R}^{m\times m}$ is the noise covariance. The measurement noises ε_k are mutually independent across measurement times and independent of the state process. When $m \neq d$, we use a linear embedding $L : \mathbb{R}^m \to \mathbb{R}^d$ that preserves the measurement information, i.e., $\sigma(LZ_k) = \sigma(Z_k)$, where $\sigma(\cdot)$ denotes the σ-algebra generated by a random variable. Under this embedding, all variables are expressed in d dimensions, with Z_k, H, R denoting $LZ_k, L \circ H, LRL^\mathsf{T}$, respectively.

Given measurements $\mathbf{z}_{1:k} = (\mathbf{z}_1, \ldots, \mathbf{z}_k)$ up to time τ_k, Bayesian filtering aims to estimate the conditional density $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$ of the state X_k [6]. This posterior distribution is computed recursively by alternating time-update and measurement-update steps.

Time-update step. Given the posterior distribution $p(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1})$ at time τ_{k-1}, the state equation (1) induces a transition distribution from X_{k-1} to X_k:

$$p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) := p(X_k = \mathbf{x}_k \,|\, X_{k-1} = \mathbf{x}_{k-1})\,. \tag{3}$$

The prior distribution $p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1})$ at time τ_k is obtained by the Chapman–Kolmogorov equation [28]:

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1})\, p(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1})\, d\mathbf{x}_{k-1}. \tag{4}$$

For $g = 0$, the dynamics are deterministic, and the transition distribution is a Dirac measure at the next state. In either case, we obtain prior samples by numerically propagating the previous posterior ensemble under the state equation over $[\tau_{k-1}, \tau_k]$.

Measurement-update step. Given the prior distribution $p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1})$, the new measurement $\mathbf{z}_k$ is incorporated through the likelihood $p(\mathbf{z}_k \mid \mathbf{x}_k)$. Under the measurement equation (2), this likelihood is given by

$$p(\mathbf{z}_k \mid \mathbf{x}_k) = \mathcal{N}(\mathbf{z}_k; H(\mathbf{x}_k), R).$$

Combining the prior distribution with this likelihood, Bayes' rule [6] gives the posterior distribution:

$$\underbrace{p(\mathbf{x}_k \mid \mathbf{z}_{1:k})}_{\text{Posterior}} \;\propto\; \underbrace{p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1})}_{\text{Prior}}\, \underbrace{p(\mathbf{z}_k \mid \mathbf{x}_k)}_{\text{Likelihood}}. \tag{5}$$

Taking the gradient of the log-posterior $\log p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$ with respect to $\mathbf{x}_k$ yields the additive score decomposition:

$$\nabla_{\mathbf{x}_k} \log p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) = \nabla_{\mathbf{x}_k} \log p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}) + \nabla_{\mathbf{x}_k} \log p(\mathbf{z}_k \mid \mathbf{x}_k). \tag{6}$$

We refer to the gradients of the log-posterior, log-prior, and log-likelihood as the posterior score, prior score, and likelihood score, respectively.

2.2 Score-based generative models

We consider a linear SDE commonly used in score-based generative modeling [24], with diffusion time $t \in [0, 1]$:

$$dX_t = F(t)X_t\,dt + G(t)\,dB_t, \qquad t \in [0, 1], \tag{7}$$

where $F(t) \in \mathbb{R}^{d\times d}$ and $G(t) \in \mathbb{R}^{d\times d}$ denote the drift and diffusion terms, respectively. A widely used instance of (7) is the variance-preserving (VP) SDE, defined by

$$F(t) = -\tfrac{1}{2}\beta(t)I_d, \qquad G(t) = \sqrt{\beta(t)}\,I_d, \tag{8}$$

where $\beta : [0, 1] \to \mathbb{R}_{\geq 0}$ is an increasing noise schedule [24], and I_d is the $d \times d$ identity matrix. For the VP coefficients in (8), the solution of (7) has the closed-form representation

$$X_t = a(t)X_0 + \gamma(t)\,\eta, \tag{9}$$

where $\eta \sim \mathcal{N}(0, I_d)$ and

$$\frac{d}{dt}\log a(t) = -\tfrac{1}{2}\beta(t), \qquad \gamma^2(t) = 1 - a^2(t). \tag{10}$$

The representation gives an explicit expression for the conditional score $\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t \mid \mathbf{x}_0)$. This expression allows score models to be trained efficiently even in high-dimensional settings via the denoising score-matching objective [26, 27].

3 Methods

3.1 Measurement-aware forward process

We construct a measurement-aware forward process that gradually transforms states from the current prior $p_0 = p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1})$ toward the measurement space. The process starts from $X_0 \sim p_0$ and evolves over diffusion time $t \in [0, 1]$, distinct from the physical time τ in (1). For d-dimensional measurements with isotropic noise covariance $R = \sigma^2 I_d$, we choose a decreasing schedule $a : [0, 1] \to [0, 1]$ satisfying $a(0) = 1$ and $a(1) = 0$, and set $\Sigma(t) = \sigma^2(1 - a^2(t))I_d$, so that $\Sigma(0) = 0$ and $\Sigma(1) = \sigma^2 I_d$. See Table 4 in Appendix A for notation.

Define an interpolation map $h : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$ by

$$h(\mathbf{x}, t) = a(t)\mathbf{x} + \big(1 - a(t)\big)H(\mathbf{x}). \tag{11}$$

Using this map, we define the measurement-aware forward process as

$$X_t = h(X_0, t) + \sqrt{\Sigma(t)}\,\eta, \qquad \eta \sim \mathcal{N}(0, I_d), \tag{12}$$

where η is independent of X_0. At the endpoints, the process starts from the state and terminates in distribution at the measurement:

$$X_0 \sim p_0, \qquad X_1 = H(X_0) + \sigma\eta \overset{d}{=} Z, \tag{13}$$

where $\overset{d}{=}$ denotes equality in distribution. We next derive a corresponding SDE whose marginal distributions match those of the forward process.

Theorem 3.1 (Non-Markovian SDE representation)**.** *Assume that $a(t)$ and $\Sigma(t)$ are differentiable, and that $\dot{\Sigma}(t)$ is positive semidefinite. Let $\widetilde{X}_0 \overset{d}{=} X_0$ be independent of the Brownian motion B_t. Suppose that $\widetilde{X}_t$ satisfies*

$$d\widetilde{X}_t = \partial_t h(\widetilde{X}_0, t)\,dt + \sqrt{\dot{\Sigma}(t)}\,dB_t, \qquad t \in [0, 1]. \tag{14}$$

Then $\mathcal{L}(\widetilde{X}_t) = \mathcal{L}(X_t)$ for all $t \in [0, 1]$, where $\mathcal{L}(Y)$ denotes the probability distribution of a random variable Y. The proof is in Appendix A.1.

Theorem 3.1 shows that the solution of the SDE has the same marginal distributions as the forward process (12). However, the process in (14) is generally non-Markovian because its drift depends on the initial variable $\widetilde{X}_0$. This dependence prevents us from applying Anderson's reverse-time theorem [29]. To address this, we apply the Markovian projection theorem [30] to the SDE in (14).

Theorem 3.2 (Markovian projection)**.** *Let $\widetilde{X}_t$ be defined by* (14)*, and assume that the regularity and integrability conditions required for the Markovian projection hold. Define*

$$b(\mathbf{x}, t) := \mathbb{E}\Big[\partial_t h(\widetilde{X}_0, t) \mid \widetilde{X}_t = \mathbf{x}\Big]. \tag{15}$$

Consider the Markovian SDE

$$d\bar{X}_t = b(\bar{X}_t, t)\, dt + \sqrt{\dot{\Sigma}(t)}\, dB_t, \qquad \bar{X}_0 \overset{d}{=} X_0, \tag{16}$$

with $\bar{X}_0$ independent of the Brownian motion B_t. Then $\mathcal{L}(\bar{X}_t) = \mathcal{L}(\widetilde{X}_t) = \mathcal{L}(X_t),\ t \in [0, 1]$. The proof is in Appendix A.2.

3.2 Estimation of the likelihood score

Theorem 3.2 provides a Markovian SDE with the same marginal distributions as the measurement-aware forward process (12). Its endpoint transition distribution will serve as the basis for estimating the likelihood score, but this requires an explicit form of the projected drift in (15). We obtain such a representation through the following two conditional means:

$$m_1(\mathbf{x}, t) := \mathbb{E}\Big[\widetilde{X}_0 \mid \widetilde{X}_t = \mathbf{x}\Big], \quad m_2(\mathbf{x}, t) := \mathbb{E}\Big[H(\widetilde{X}_0) \mid \widetilde{X}_t = \mathbf{x}\Big]. \tag{17}$$

Here, $m_1(\mathbf{x}, t)$ estimates the initial state associated with the perturbed state $\mathbf{x}$, whereas $m_2(\mathbf{x}, t)$ estimates the corresponding clean measurement. These two conditional means determine the drift as

$$b(\mathbf{x}, t) = \dot{a}(t)\,(m_1(\mathbf{x}, t) - m_2(\mathbf{x}, t)). \tag{18}$$

Because this drift is generally nonlinear, the likelihood $p(\mathbf{z} \mid \mathbf{x}_t)$ cannot be evaluated in closed form. We therefore approximate the endpoint transition distribution $p(\mathbf{z} \mid \mathbf{x}_t)$ of the projected SDE (16). Specifically, we integrate the projected SDE from t to 1:

$$\bar{X}_1 = \bar{X}_t + \int_t^1 \dot{a}(u)\,\big(m_1(\bar{X}_u, u) - m_2(\bar{X}_u, u)\big)\, du + \int_t^1 \sqrt{\dot{\Sigma}(u)}\, dB_u. \tag{19}$$

Approximating the integral by freezing the conditional means at $(\bar{X}_t, t)$ yields

$$\bar{X}_1 \approx \bar{X}_t - a(t)\,\big(m_1(\bar{X}_t, t) - m_2(\bar{X}_t, t)\big) + \sqrt{\Sigma_{t\to 1}}\,\eta, \qquad \eta \sim \mathcal{N}(0, I), \tag{20}$$

where $\Sigma_{s\to t} := \Sigma(t) - \Sigma(s)$.

Proposition 3.3 (Endpoint approximation for the likelihood score)**.** *For $0 \le t < 1$, define*

$$\mu_t(\mathbf{x}) := \mathbf{x} - a(t)\,(m_1(\mathbf{x}, t) - m_2(\mathbf{x}, t)). \tag{21}$$

Assume that μ_t is differentiable with respect to $\mathbf{x}$ and that $\Sigma_{t\to 1}$ is positive definite. Under the endpoint approximation in (20)*, the conditional likelihood is approximated by*

$$p(\mathbf{z} \mid \mathbf{x}_t = \mathbf{x}) \approx \mathcal{N}\left(\mathbf{z}; \mu_t(\mathbf{x}), \Sigma_{t\to 1}\right). \tag{22}$$

Consequently, the corresponding likelihood-score approximation is

$$\begin{aligned} g(\mathbf{z}, \mathbf{x}, t) &:= \nabla_{\mathbf{x}} \log \mathcal{N}\left(\mathbf{z}; \mu_t(\mathbf{x}), \Sigma_{t\to 1}\right) \\ &= \left(\nabla_{\mathbf{x}} \mu_t(\mathbf{x})\right)^{\mathsf{T}} \Sigma_{t\to 1}^{-1}\left(\mathbf{z} - \mu_t(\mathbf{x})\right). \end{aligned} \tag{23}$$

The derivation is in Appendix A.3.

Proposition 3.3 applies to general nonlinear measurement operators. For linear measurements, we construct a linear SDE with the same one-time marginal distributions and derive an exact likelihood score.

Theorem 3.4 (Exact likelihood score for linear measurements)**.** *Suppose that $H(\mathbf{x}) = A\mathbf{x}$ for some matrix $A \in \mathbb{R}^{d\times d}$, and define*

$$A(t) = a(t)I + \big(1 - a(t)\big)A. \tag{24}$$

Assume that $A(t)$ is invertible for $t < 1$, and define $F_t = \dot{A}(t)A(t)^{-1}$. Let G_t satisfy

$$G_t G_t^{\mathsf{T}} = \dot{\Sigma}(t) - F_t\Sigma(t) - \Sigma(t)F_t^{\mathsf{T}} \succeq 0. \tag{25}$$

Consider the linear SDE $d\check{X}_t = F_t\check{X}_t\,dt + G_t\,dB_t$, initialized with $\check{X}_0 \overset{d}{=} X_0$. Assume that the associated Fokker–Planck equation is well posed for $t < 1$ and that the transition distributions have weak limits at $t = 1$. Endpoint transitions are interpreted through these limits. Then the solution of this linear SDE has the same marginal distributions as X_t and the transition distribution

$$\check{X}_t \mid \check{X}_s \sim \mathcal{N}\left(M_{s\to t}\check{X}_s,\, \check{\Sigma}_{s\to t}\right), \qquad 0 \le s < t < 1, \tag{26}$$

where

$$M_{s\to t} = A(t)A(s)^{-1}, \qquad \check{\Sigma}_{s\to t} = \Sigma(t) - M_{s\to t}\Sigma(s)M_{s\to t}^{\mathsf{T}}. \tag{27}$$

Consequently, for $0 \le t < 1$ with $\check{\Sigma}_{t\to 1}$ positive definite, the likelihood score is

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{z} \mid \mathbf{x}_t) = M_{t\to 1}^{\mathsf{T}}\check{\Sigma}_{t\to 1}^{-1}\left(\mathbf{z} - M_{t\to 1}\mathbf{x}_t\right). \tag{28}$$

The proof is in Appendix A.4.

3.3 Guided reverse-time sampling

We perform guided reverse-time sampling using the projected SDE in Theorem 3.2 for nonlinear measurements and the linear SDE in Theorem 3.4 for linear measurements. The reverse updates combine the prior score with the corresponding likelihood score $g(\mathbf{z}, \mathbf{x}, t)$, given by (23) or (28).

Lemma 3.5 (Guided reverse update)**.** *For $t < s < 1$ and a given state $\mathbf{x}_s$, define*

$$\widehat{h}_s(\mathbf{x}_s) := a(s)m_1(\mathbf{x}_s, s) + \big(1 - a(s)\big)m_2(\mathbf{x}_s, s). \tag{29}$$

Using Tweedie's formula [31], the prior score at s is estimated as

$$s_{\text{prior}}(\mathbf{x}_s, s) := -\Sigma(s)^{-1}\big(\mathbf{x}_s - \widehat{h}_s(\mathbf{x}_s)\big). \tag{30}$$

Based on the posterior-score decomposition in (6)*, we approximate the posterior score by combining the prior and likelihood scores:*

$$s_{\text{guide}}(\mathbf{x}_s, s, \mathbf{z}) := s_{\text{prior}}(\mathbf{x}_s, s) + g(\mathbf{z}, \mathbf{x}_s, s). \tag{31}$$

For nonlinear measurements, discretizing the projected SDE (16) *gives the update, with $\eta \sim \mathcal{N}(0, I)$:*

$$\mathbf{x}_t = \mathbf{x}_s + \big(a(t) - a(s)\big)\big(m_1(\mathbf{x}_s, s) - m_2(\mathbf{x}_s, s)\big) - \Sigma_{s\to t}s_{\text{guide}}(\mathbf{x}_s, s, \mathbf{z}) + \sqrt{-\Sigma_{s\to t}}\,\eta. \tag{32}$$

For linear measurements, we use the transition distribution in (26)*, with the parameters in* (27) *evaluated for $s > t$:*

$$\mathbf{x}_t = M_{s\to t}\mathbf{x}_s - \check{\Sigma}_{s\to t}s_{\text{guide}}(\mathbf{x}_s, s, \mathbf{z}) + \sqrt{-\check{\Sigma}_{s\to t}}\,\eta. \tag{33}$$

With $-\Sigma_{s\to t} \succeq 0$ and $-\check{\Sigma}_{s\to t} \succeq 0$, both updates are finite-step guided approximations. The derivation is in Appendix A.5.

To compute these updates, we estimate the conditional means m_1 and m_2 from samples of the forward process in (12) using denoising L_2 objectives.

Proposition 3.6 (Denoising objective for conditional means)**.** *Let $m_{1,\theta}$ and $m_{2,\theta}$ be parametric estimators of m_1 and m_2 in* (17)*. Let $y_1(\mathbf{x}_0) = \mathbf{x}_0$ and $y_2(\mathbf{x}_0) = H(\mathbf{x}_0)$, and define*

$$L_i(\theta, t) := \mathbb{E}_{\mathbf{x}_t\sim\tilde{p}_t}\left[\|m_{i,\theta}(\mathbf{x}_t, t) - m_i(\mathbf{x}_t, t)\|^2\right], \tag{34}$$

$$\widehat{L}_i(\theta, t) := \mathbb{E}_{\mathbf{x}_0\sim p_0}\mathbb{E}_{\mathbf{x}_t\sim p_t(\cdot|\mathbf{x}_0)}\left[\|m_{i,\theta}(\mathbf{x}_t, t) - y_i(\mathbf{x}_0)\|^2\right], \tag{35}$$

where $i = 1, 2$, $\tilde{p}_t$ is the marginal distribution of $\widetilde{X}_t$ in (14) *and $p_t(\cdot \mid \mathbf{x}_0)$ is the conditional distribution of X_t in* (12)*. Then*

$$\nabla_\theta L_i(\theta, t) = \nabla_\theta \widehat{L}_i(\theta, t), \qquad i = 1, 2. \tag{36}$$

Thus, we train $m_{1,\theta}$ and $m_{2,\theta}$ by minimizing $\mathcal{L}_{\text{reg}}(\theta, t) = \widehat{L}_1(\theta, t) + \widehat{L}_2(\theta, t)$. For linear measurements $H(\mathbf{x}) = A\mathbf{x}$, linearity of conditional expectation gives $m_2 = Am_1$, so we minimize only $\widehat{L}_1(\theta, t)$ and set $m_{2,\theta} = Am_{1,\theta}$. The proof is in Appendix A.6.

Method	RMSE (↓)	fCRPS (↓)	SSR (→ 1)
EnKF	0.346 ± 0.070	0.103 ± 0.025	0.720 ± 0.122
LETKF	0.281 ± 0.044	0.081 ± 0.009	0.938 ± 0.141
SF	1.089 ± 0.221	0.733 ± 0.145	0.315 ± 0.083
SSLS	0.235 ± 0.025	0.156 ± 0.004	3.685 ± 0.413
MASF	$\mathbf{0.081 \pm 0.003}$	$\mathbf{0.044 \pm 0.001}$	$\mathbf{0.997 \pm 0.035}$

Table 1: **Lorenz96 with nonlinear cubic measurements.** All methods use $N = 100$ ensemble members. Mean $\pm$ standard deviation over 20 seeds.

3.4 Pretraining and filtering

Pretraining. For offline pretraining, we draw a set of samples from the initial distribution and propagate them under the state equation to the first measurement time. These prior samples are perturbed at diffusion times using (12), and the conditional-mean estimators are trained to estimate the original state and, for nonlinear measurements, the clean measurement using the denoising objectives in (35).

Filtering. We initialize the prior ensemble with N samples from the prior distribution at the first measurement time. At each measurement time, we fine-tune the estimators on this ensemble perturbed using (12), with the denoising objectives in (35). These updates start from pretrained weights initially and previous-step weights thereafter. The updated estimators provide the prior score through (30), which is combined with the likelihood score for the current measurement in (31) to guide reverse sampling via (32) or (33). Sampling starts from the prior ensemble perturbed near $t = 1$ and proceeds toward $t = 0$, yielding an ensemble whose mean provides the state estimate. This ensemble is propagated under the state equation to form the prior ensemble at the next measurement time.

Algorithm 1 in Appendix B summarizes the procedure, and Appendix C provides additional implementation details on the architecture, optimization settings, and noise schedule.

4 Experiments

We compare MASF with EnKF, LETKF, SF, and SSLS on Lorenz96 and Kolmogorov flow. The 40-dimensional Lorenz96 system serves as a sanity check for nonlinear measurements, while Kolmogorov flow tests accuracy and online wall-clock efficiency in high dimensions across different measurement settings. Performance is assessed using root mean squared error (RMSE), fair continuous ranked probability score (fCRPS), and spread–skill ratio (SSR). See Appendix C.1 for metric definitions.

4.1 Lorenz96

Lorenz96 is an idealized model of atmospheric dynamics that captures nonlinear interactions among variables [32]. We first compare the methods under nonlinear measurements using the 40-dimensional Lorenz96 system governed by

$$\frac{dx_i}{d\tau} = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F, \qquad F = 8, \quad i = 1, \dots, 40, \tag{37}$$

with periodic indices $x_{i+40} = x_i$. We integrate the drift with fourth-order Runge–Kutta using $\Delta\tau = 0.05$, adding independent process noise with standard deviation $\sigma_p\sqrt{\Delta\tau}$ at each step, where $\sigma_p = 0.1$. The initial prior is $\mathcal{N}(0, I_{40})$, and the first 25 steps form the spin-up period.

Measurement operators. All components are measured through the monotonic cubic operator

$$[H(\mathbf{x})]_i = x_i + 0.01x_i^3, \qquad i = 1, \dots, 40. \tag{38}$$

Measurements include additive noise with standard deviation $\sigma = 0.1$ and are assimilated at simulation steps $25, 35, \dots, 125$. All methods use 100 ensemble members, and results are averaged over 20 random seeds. In this nonlinear measurement setting, MASF achieves the lowest RMSE of 0.081 and fCRPS of 0.044, with an SSR of 0.997 indicating agreement between ensemble spread and estimation error (Table 1). Further simulation details and settings are provided in Appendix C.2.

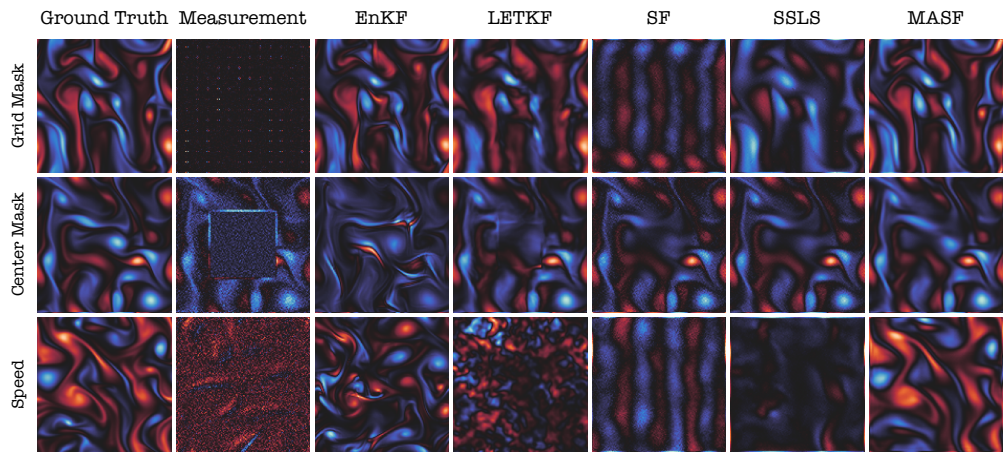


Figure 2: **Qualitative results.** Rows show grid-mask, center-mask, and speed measurements, from top to bottom. Estimates are ensemble means.

4.2 Kolmogorov flow

Kolmogorov flow is a two-dimensional incompressible flow driven by external forcing. Its velocity field $\mathbf{u}$ follows the forced incompressible Navier–Stokes equations:

$$\partial_\tau \mathbf{u} = -(\mathbf{u}\cdot\nabla)\mathbf{u} + \frac{1}{\mathrm{Re}}\nabla^2\mathbf{u} - \frac{1}{\rho}\nabla p + \mathbf{f}, \qquad \nabla\cdot\mathbf{u} = 0. \tag{39}$$

Here, p denotes pressure and $\mathbf{f}$ the external forcing. We simulate the system on $[0, 2\pi]^2$ using JAX-CFD [33], with density $\rho = 1$, Reynolds number $\mathrm{Re} = 1000$, and a time interval of $\Delta\tau = 0.2$ between successive states. For the main comparisons, we discretize the two velocity components on a 128×128 grid. Unless otherwise stated, measurements are taken every 10 steps over steps 50–100, with additive noise with standard deviation $\sigma = 0.1$.

Measurement operators. For Kolmogorov flow, the main comparisons use grid masks, center masks, and a speed operator. The following spatial masks are applied element-wise to both velocity channels:

$$\begin{aligned}[A_{\mathrm{grid}}]_{ij} &= \mathbf{1}\{i \equiv 0 \pmod s,\ j \equiv 0 \pmod s\},\\ [A_{\mathrm{center}}]_{ij} &= \mathbf{1}\{(i,j) \notin \mathcal{C}_h\},\end{aligned} \tag{40}$$

where $s = 10$ at 128^2 resolution and $\mathcal{C}_h$ is a central square of side length $h = 64$. The speed operator transforms the squared speed and repeats the resulting scalar in both measurement channels:

$$[H(\mathbf{x})]_{1ij} = [H(\mathbf{x})]_{2ij} = \mathrm{sigmoid}\left(\sum_{c=1}^{2} x_{cij}^2 - \kappa\right). \tag{41}$$

Here, κ is a shift parameter, and independent noise is added after channel replication. For these linear and nonlinear measurement settings, we use a U-Net and a dual-head U-Net, respectively [34, 35], with SF and SSLS sharing the same backbone where applicable.

Fig. 2 provides qualitative comparisons of the state estimates under the three measurements and Table 2 shows that MASF achieves the lowest RMSE and fCRPS across all three measurements. For grid-mask measurements, EnKF and LETKF achieve RMSE close to that of MASF, whereas SF and SSLS have substantially higher RMSE. Center-mask measurements are less sparse than grid-mask measurements, and SF and SSLS achieve lower RMSE than with grid masks, while EnKF's RMSE increases sharply. LETKF also becomes substantially slower because the average number of measurements within each localization window increases from 6.5 to 475.5.

The RMSE gap between MASF and the best baseline is largest for speed measurements. Among the four baselines, SSLS has the lowest RMSE, but both its RMSE and fCRPS are approximately three times those of MASF. In this setting, MASF also achieves a $20.7\times$ online wall-clock speedup over LETKF. Alongside these reductions in RMSE, MASF's SSR is close to one for both grid-mask and speed measurements, indicating agreement between ensemble spread and error.

Measurement	Method	N	RMSE ($\downarrow$)	fCRPS ($\downarrow$)	SSR ($\rightarrow 1$)	Time (s)	Speedup
Grid mask	EnKF	250	0.18 ± 0.04	0.09 ± 0.03	0.56 ± 0.09	180.3 ± 0.4	$1.5\times$
	LETKF	40	0.17 ± 0.01	0.09 ± 0.01	1.44 ± 0.09	389.6 ± 5.4	$3.3\times$
	SF	100	0.81 ± 0.03	0.45 ± 0.02	0.91 ± 0.03	670.6 ± 0.9	$\mathbf{5.6\times}$
	SSLS	100	0.51 ± 0.03	0.27 ± 0.02	0.97 ± 0.05	611.5 ± 1.1	$5.1\times$
	MASF	10	$\mathbf{0.15 \pm 0.02}$	$\mathbf{0.07 \pm 0.01}$	$\mathbf{1.01 \pm 0.05}$	$\mathbf{118.8 \pm 2.5}$	–
Center mask	EnKF	250	0.75 ± 0.17	0.52 ± 0.13	0.03 ± 0.01	185.4 ± 2.5	$1.1\times$
	LETKF	40	0.34 ± 0.05	0.14 ± 0.03	$\mathbf{1.02 \pm 0.22}$	2294.1 ± 45.2	$\mathbf{14.0\times}$
	SF	100	0.29 ± 0.03	0.13 ± 0.02	0.80 ± 0.06	672.3 ± 3.4	$4.1\times$
	SSLS	100	0.29 ± 0.05	0.11 ± 0.02	0.80 ± 0.17	610.8 ± 0.8	$3.7\times$
	MASF	10	$\mathbf{0.18 \pm 0.04}$	$\mathbf{0.07 \pm 0.02}$	0.78 ± 0.19	$\mathbf{164.1 \pm 1.9}$	–
Speed	EnKF	250	1.29 ± 0.05	1.01 ± 0.05	0.04 ± 0.01	190.7 ± 0.7	$1.5\times$
	LETKF	40	0.90 ± 0.13	0.51 ± 0.09	0.68 ± 0.10	2567.6 ± 9.6	$\mathbf{20.7\times}$
	SF	100	0.89 ± 0.03	0.49 ± 0.02	0.92 ± 0.03	380.5 ± 7.1	$3.1\times$
	SSLS	100	0.79 ± 0.05	0.41 ± 0.03	1.09 ± 0.03	358.9 ± 2.2	$2.9\times$
	MASF	10	$\mathbf{0.27 \pm 0.04}$	$\mathbf{0.14 \pm 0.01}$	$\mathbf{0.92 \pm 0.11}$	$\mathbf{123.9 \pm 1.0}$	–

Table 2: **Kolmogorov flow at 128^2 resolution.** Results for grid-mask, center-mask, and nonlinear speed measurements. Mean $\pm$ standard deviation over five seeds. N is the ensemble size, and time denotes online wall-clock time. Speedup is the ratio of each baseline's time to MASF's for the same measurement.

An additional baseline comparison, experiments with shifted Reynolds numbers, and qualitative comparisons at additional time steps are provided in Appendices D.1, D.2, and D.3, respectively.

4.3 Scaling to high resolution

We assess scalability to $\mathcal{O}(10^5)$ dimensions using grid-mask measurements at 256^2 and 512^2 resolutions. The grid strides are 15 and 20, respectively, compared with 10 at 128^2. Table 3 shows that at 256^2, MASF achieves the lowest RMSE while being $3.0\times$–$4.8\times$ faster than the baselines. At 512^2, MASF remains stable and achieves the lowest RMSE while being $10.6\times$ faster than EnKF and $3.9\times$ faster than LETKF. SF and SSLS are omitted at this resolution due to numerical instability.

Resolution	Method	N	RMSE ($\downarrow$)	Time (s)	Speedup
256^2	EnKF	250	0.17 ± 0.03	1276.3 ± 5.8	$3.7\times$
	LETKF	40	0.19 ± 0.01	1646.7 ± 15.2	$\mathbf{4.8\times}$
	SF	100	0.88 ± 0.03	1056.2 ± 4.3	$3.1\times$
	SSLS	100	0.72 ± 0.06	1016.7 ± 8.8	$3.0\times$
	MASF	10	$\mathbf{0.16 \pm 0.02}$	$\mathbf{343.2 \pm 4.0}$	-
512^2	EnKF	250	0.18 ± 0.07	11162.5 ± 80.3	$\mathbf{10.6\times}$
	LETKF	40	0.30 ± 0.02	4140.1 ± 37.4	$3.9\times$
	MASF	10	$\mathbf{0.13 \pm 0.01}$	$\mathbf{1049.4 \pm 22.1}$	-

Table 3: **High-dimensional scaling of Kolmogorov flow on grid-mask measurements.** Mean $\pm$ standard deviation over five seeds. Time denotes online wall-clock time. Speedup is the ratio of each baseline's time to MASF's at the same resolution.

4.4 Wall-clock time, accuracy, and sensitivity analysis

Fig. 3 summarizes RMSE, wall-clock time, and sensitivity under grid-mask measurements, omitting SF for visual clarity because of its larger RMSE. MASF consistently achieves low RMSE with substantially shorter wall-clock time than SSLS. EnKF reduces its RMSE with larger ensembles, while LETKF achieves RMSE close to MASF's. Both require more wall-clock time, and EnKF shows higher RMSE variability.

With the measurement interval fixed at 15 simulation steps, MASF maintains the lowest RMSE as the final measurement time increases, while EnKF's RMSE rises and SSLS's gradually declines.

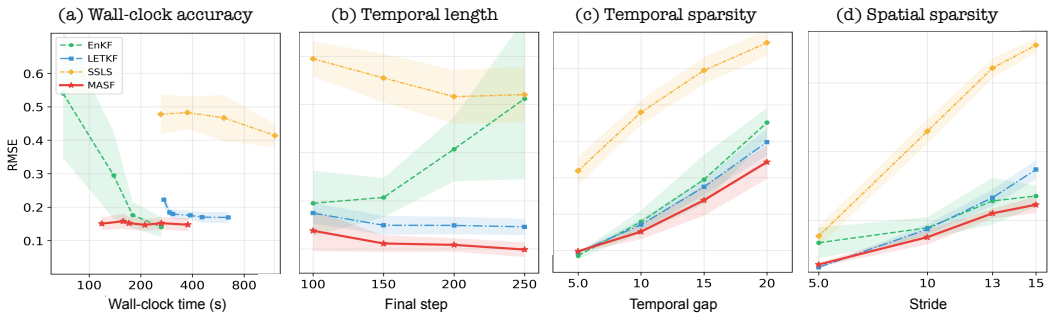


Figure 3: **Wall-clock time, accuracy, and sensitivity analysis.** Comparison under grid-mask measurements across wall-clock time, temporal length, temporal gap, and spatial sparsity. Shaded regions denote standard deviation over five seeds.

This advantage persists as temporal and spatial sparsity increase, with MASF's RMSE rising more slowly than those of the baselines. In contrast, SSLS has RMSE comparable to EnKF at grid-mask stride 5, but its RMSE rises sharply from stride 10. Additional ensemble-size and sensitivity results appear in Appendices D.4 and D.5, and component and pretraining ablations in Appendices E.1 and E.2. Reported MASF wall-clock time includes online fine-tuning and sampling but excludes offline pretraining, which is included in the amortized-cost analysis in Appendix E.3.

5 Limitations and broader impacts

Limitations. MASF assumes known state and measurement equations, and may degrade under misspecification or extremely noisy measurements. For nonlinear measurements, it relies on an endpoint approximation. We have not established theoretical convergence of reverse sampling to the posterior distribution. MASF also requires online fine-tuning and depends on hyperparameters such as normalization and fine-tuning epochs. Our experiments focus on Lorenz96 and Kolmogorov flow, with the latter evaluated over five seeds at resolutions up to 512^2. Broader systems, larger 3D domains, and more extensive evaluation remain future work.

Broader impacts. MASF may support fluid monitoring, forecasting, and sensor-based control by reducing online filtering cost. Model errors, biased measurements, and overconfident predictions can nevertheless yield unreliable decisions, requiring validation and uncertainty assessment in high-stakes applications.

6 Conclusion

We propose the Measurement-Aware Score-Based Filter (MASF), which incorporates the measurement equation into the forward process for score-based data assimilation. From this forward process, we derive a projected SDE with matching marginal distributions and use its endpoint transition distribution to obtain a theoretically grounded likelihood-score approximation for nonlinear measurements. For linear measurements, we derive an exact likelihood score using a linear SDE with the same marginal distributions. Experiments on Lorenz96 and Kolmogorov flow show that MASF achieves the lowest RMSE and fCRPS among the evaluated methods in the main benchmark. With amortized pretraining and a lightweight model, MASF also improves online wall-clock efficiency on Kolmogorov flow, while remaining effective at state dimensions exceeding 10^5. These results position MASF as a framework for score-based data assimilation that combines efficiency and scalability in nonlinear high-dimensional systems.

Acknowledgments and Disclosure of Funding

We thank Won Chang (Department of Statistics, Seoul National University) for helpful advice on amortized pretraining and the discussion of Bayesian filtering. We also thank Woo-Joo Kim (School of Electrical Engineering, KAIST) for help with figure revisions and writing, and for encouragement.

This work was supported by grants to E. Yoon from the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (Grant No. RS-2026-25572974).

This work was supported by grants to D. Kim from the National Research Foundation of Korea under Grant RS-2025-00513609, and by an Ewon Assistant Professorship Award at KAIST.

This work was supported by grants to D. W. Kim from the National Research Foundation of Korea (Grant Nos. RS-2025-00561696, RS-2025-02215354, RS-2026-25522728, RS-2026-25512726, RS-2026-25609011, and RS-2026-25613293), the Korea Basic Science Institute (Grant No. RS-2026-25500300), Kaifo Holdings Co., Ltd., the Korean subsidiary of Formosa Biotechnology Co., Ltd., a member of the Formosa Group, Taiwan (Grant No. G01260187), and the KAIST Start-up Research Fund (Grant No. G04240060).

References

[1] Geir Evensen. *Data Assimilation: The Ensemble Kalman Filter*. Springer Berlin Heidelberg, 2nd edition, 2009. doi: 10.1007/978-3-642-03711-5.

[2] Sebastian Reich and Colin Cotter. *Probabilistic Forecasting and Bayesian Data Assimilation*. Cambridge University Press, 2015.

[3] Hristo G. Chipilski, Xuguang Wang, and David B. Parsons. Impact of assimilating pecan profilers on the prediction of bore-driven nocturnal convection: A multiscale forecast evaluation for the 6 july 2015 case study. *Monthly Weather Review*, 148(3):1147 – 1175, 2020. doi: 10.1175/MWR-D-19-0171.1. URL https://journals.ametsoc.org/view/journals/mwre/148/3/mwr-d-19-0171.1.xml.

[4] Altuğ Aksoy, David C. Dowell, and Chris Snyder. A multicase comparative assessment of the ensemble kalman filter for assimilation of radar observations. part i: Storm-scale analyses. *Monthly Weather Review*, 137(6):1805 – 1824, 2009. doi: 10.1175/2008MWR2691.1. URL https://journals.ametsoc.org/view/journals/mwre/137/6/2008mwr2691.1.xml.

[5] N. G. Cogan, Feng Bao, Ralf Paus, and Atanaska Dobreva. Data assimilation of synthetic data as a novel strategy for predicting disease progression in alopecia areata. *Mathematical Medicine and Biology*, 38(3):314–332, September 2021. ISSN 1477-8599. doi: 10.1093/imammb/dqab008.

[6] Simo Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013. doi: 10.1017/CBO9781139344203.

[7] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960. ISSN 0021-9223. doi: 10.1115/1.3662552.

[8] Geir Evensen. The ensemble kalman filter for combined state and parameter estimation. *IEEE Control Systems Magazine*, 29(3):83–104, 2009. doi: 10.1109/MCS.2009.932223.

[9] Jeffrey S. Whitaker and Thomas M. Hamill. Ensemble data assimilation without perturbed observations. *Monthly Weather Review*, 130(7):1913 – 1924, 2002. doi: 10.1175/1520-0493(2002)130<1913:EDAWPO>2.0.CO;2. URL https://journals.ametsoc.org/view/journals/mwre/130/7/1520-0493_2002_130_1913_edawpo_2.0.co_2.xml.

[10] Brian R. Hunt, Eric J. Kostelich, and Istvan Szunyogh. Efficient data assimilation for spatiotemporal chaos: A local ensemble transform kalman filter. *Physica D: Nonlinear Phenomena*, 230(1):112–126, 2007. ISSN 0167-2789. doi: 10.1016/j.physd.2006.11.008. URL https://www.sciencedirect.com/science/article/pii/S0167278906004647.

[11] P. L. Houtekamer and Herschel L. Mitchell. Data assimilation using an ensemble kalman filter technique. *Monthly Weather Review*, 126(3):796 – 811, 1998. doi: 10.1175/1520-0493(1998)126<0796:DAUAEK>2.0.CO;2. URL https://journals.ametsoc.org/view/journals/mwre/126/3/1520-0493_1998_126_0796_dauaek_2.0.co_2.xml.

[12] Alessio Spantini, Ricardo Baptista, and Youssef Marzouk. Coupling techniques for nonlinear ensemble filtering. *SIAM Review*, 64(4):921–953, 2022. doi: 10.1137/20M1312204.

[13] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140:107–113, 1993. doi: 10.1049/ip-f-2.1993.0015. URL https://digital-library.theiet.org/doi/abs/10.1049/ip-f-2.1993.0015.

[14] Michael K. Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999. doi: 10.1080/01621459.1999.10474153. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1999.10474153.

[15] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002. doi: 10.1109/78.978374.

[16] Chris Snyder, Thomas Bengtsson, Peter Bickel, and Jeff Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629 – 4640, 2008. doi: 10.1175/2008MWR2529.1. URL https://journals.ametsoc.org/view/journals/mwre/136/12/2008mwr2529.1.xml.

[17] Mohammad Al-Jarrah, Bamdad Hosseini, and Amirhossein Taghvaei. Optimal transport particle filters. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 6798–6805, 2023. doi: 10.1109/CDC49753.2023.10383337.

[18] Mohammad Al-Jarrah, Niyizhen Jin, Bamdad Hosseini, and Amirhossein Taghvaei. Nonlinear filtering with brenier optimal transport maps. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 813–839. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/al-jarrah24a.html.

[19] François Rozet and Gilles Louppe. Score-based data assimilation. In *Advances in Neural Information Processing Systems*, volume 36, pages 40521–40541. Curran Associates, Inc., 2023. doi: 10.52202/075280-1763. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/7f7fa581cc8a1970a4332920cdf87395-Paper-Conference.pdf.

[20] Feng Bao, Zezhong Zhang, and Guannan Zhang. A score-based filter for nonlinear data assimilation. *Journal of Computational Physics*, 514:113207, 2024. ISSN 0021-9991. doi: 10.1016/j.jcp.2024.113207. URL https://www.sciencedirect.com/science/article/pii/S002199912400456X.

[21] 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. ISSN 0045-7825. doi: 10.1016/j.cma.2024.117447. URL https://www.sciencedirect.com/science/article/pii/S0045782524007023.

[22] Zhao Ding, Chenguang Duan, Yuling Jiao, Jerry Zhijian Yang, Cheng Yuan, and Pingwen Zhang. Nonlinear assimilation via score-based sequential langevin sampling. *IEEE Transactions on Information Theory*, 72(10):7754–7784, 2026. doi: 10.1109/TIT.2026.3715654.

[23] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/3001ef257407d5a371a96dcd947c7d93-Paper.pdf.

[24] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

[25] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf.

[26] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *J. Mach. Learn. Res.*, 6:695–709, 2005.

[27] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142.

[28] Kody Law, Andrew Stuart, and Konstantinos Zygalakis. *Data Assimilation: A Mathematical Introduction*. Springer, 2015. doi: 10.1007/978-3-319-20325-6.

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

[30] Gerard Brunick and Steven Shreve. Mimicking an itô process by a solution of a stochastic differential equation. *The Annals of Applied Probability*, 23:1584–1628, 08 2013. doi: 10.1214/12-AAP881.

[31] Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. doi: 10.1198/jasa.2011.tm11181.

[32] Edward N. Lorenz and Kerry A. Emanuel. Optimal sites for supplementary weather observations: Simulation with a small model. *Journal of the Atmospheric Sciences*, 55(3):399 – 414, 1998. doi: 10.1175/1520-0469(1998)055<0399:OSFSWO>2.0.CO;2. URL https://journals.ametsoc.org/view/journals/atsc/55/3/1520-0469_1998_055_0399_osfswo_2.0.co_2.xml.

[33] Dmitrii Kochkov, Jamie A. Smith, Ayya Alieva, Qing Wang, Michael P. Brenner, and Stephan Hoyer. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021. doi: 10.1073/pnas.2101784118. URL https://www.pnas.org/doi/abs/10.1073/pnas.2101784118.

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

[35] Ryohei Kuga, Asako Kanezaki, Masaki Samejima, Yusuke Sugano, and Yasuyuki Matsushita. Multi-task learning using multi-modal encoder-decoder networks with shared skip connections. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 403–411, 2017. doi: 10.1109/ICCVW.2017.54.

[36] C. A. T. Ferro. Fair scores for ensemble forecasts. *Quarterly Journal of the Royal Meteorological Society*, 140(683):1917–1923, 2014. doi: 10.1002/qj.2270.

[37] V. Fortin, M. Abaza, F. Anctil, and R. Turcotte. Why should ensemble spread match the rmse of the ensemble mean? *Journal of Hydrometeorology*, 15(4):1708 – 1713, 2014. doi: 10.1175/JHM-D-14-0008.1. URL https://journals.ametsoc.org/view/journals/hydr/15/4/jhm-d-14-0008_1.xml.

[38] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171. PMLR, 18–24 Jul 2021.

[39] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2623–2631, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330701.

A Theoretical details

Notation. Table 4 summarizes the main symbols used throughout the paper.

Symbols	Description	Defined in
τ	The physical time of the dynamical system.	Eq. (1)
$t \in [0,1]$	The diffusion time of the forward process.	Sec. 3.1
X_k	The state at the measurement time τ_k.	Sec. 2.1
Z_k	The measurement at the measurement time τ_k.	Eq. (2)
$\mathbf{z}_{1:k}$	The observed measurement sequence $(\mathbf{z}_1, \dots, \mathbf{z}_k)$.	Sec. 2.1
X_t	The measurement-aware forward process.	Eq. (12)
$\widetilde{X}_t$	The solution of the non-Markovian SDE of the forward process.	Eq. (14)
$\bar{X}_t$	The solution of the Markovian SDE.	Eq. (16)
$\check{X}_t$	The solution of the linear SDE.	Thm. 3.4
H	The measurement operator.	Eq. (2)
R	The measurement-noise covariance.	Eq. (2)
$a(t)$	The interpolation schedule with $a(0) = 1$ and $a(1) = 0$.	Sec. 3.1
$h(\mathbf{x}, t)$	The interpolation map between the identity map and the measurement operator.	Eq. (11)
$\Sigma(t)$	The noise covariance schedule, $\Sigma(t) = \sigma^2(1 - a^2(t))I_d$.	Sec. 3.1
$\Sigma_{s\to t}$	The covariance increment $\Sigma_{s\to t} := \Sigma(t) - \Sigma(s)$.	Sec. 3.2
$b(\mathbf{x}, t)$	The projected Markov drift.	Eq. (15)
$m_1(\mathbf{x}, t)$	The conditional mean of the initial state.	Eq. (17)
$m_2(\mathbf{x}, t)$	The conditional mean of the clean measurement $H(\widetilde{X}_0)$.	Eq. (17)
$\mu_t(\mathbf{x})$	The mean of the endpoint approximation.	Eq. (21)
$g(\mathbf{z}, \mathbf{x}, t)$	The likelihood score.	Eqs. (23, 28)
s_{prior}	The prior score estimator.	Eq. (30)
s_{guide}	The guided posterior score estimator.	Eq. (31)
A	The matrix representation of a linear measurement operator.	Thm. 3.4
$A(t)$	The linear interpolation matrix $A(t) = a(t)I + (1 - a(t))A$.	Eq. (24)
$M_{t\to 1}$	The transition matrix from diffusion time t to the endpoint in the linear case.	Eq. (27)
$\check{\Sigma}_{t\to 1}$	The endpoint transition covariance for the linear SDE.	Eq. (27)

Table 4: **Summary of notation.**

This section develops the SDEs underlying MASF. We first define a measurement-aware forward process X_t, then construct a non-Markovian SDE $\widetilde{X}_t$ with the same marginal distributions, and obtain a Markovian SDE $\bar{X}_t$ via Markovian projection. The projected process provides the basis for likelihood-score estimation. For nonlinear measurements, we use an endpoint approximation for the likelihood score. For linear measurements, a linear SDE has the same marginal distributions as X_t, and its transition distribution gives the exact likelihood score. The resulting likelihood score is combined with the prior score to construct the guided reverse-time sampler.

A.1 Measurement-aware forward process

Let $X_0 \sim p_0$ denote a state drawn from the current prior. Let $a : [0,1] \to [0,1]$ be a decreasing schedule with $a(0) = 1$ and $a(1) = 0$, and define

$$\Sigma(t) = \sigma^2(1 - a^2(t))I_d. \tag{42}$$

For a measurement operator H with noise covariance $R = \sigma^2 I_d$, define

$$h(\mathbf{x}, t) = a(t)\mathbf{x} + \big(1 - a(t)\big)H(\mathbf{x}). \tag{43}$$

The resulting measurement-aware forward process is

$$X_t = h(X_0, t) + \sqrt{\Sigma(t)}\,\eta, \qquad \eta \sim \mathcal{N}(0, I), \qquad t \in [0,1], \tag{44}$$

where η is independent of X_0. By construction,

$$X_0 \sim p_0, \qquad X_1 = H(X_0) + \sigma\eta \overset{d}{=} Z. \tag{45}$$

Theorem A.1 (Non-Markovian SDE representation)**.** *Assume that $a(t)$ and $\Sigma(t)$ are differentiable and $\dot{\Sigma}(t)$ is positive semidefinite. Let $\widetilde{X}_0 \overset{d}{=} X_0$ be independent of the Brownian motion B_t. Define*

$$d\widetilde{X}_t = \partial_t h(\widetilde{X}_0, t)\, dt + \sqrt{\dot{\Sigma}(t)}\, dB_t, \qquad t \in [0,1]. \tag{46}$$

Then $\widetilde{X}_t$ has the same marginal distributions as the measurement-aware forward process:

$$\mathcal{L}(\widetilde{X}_t) = \mathcal{L}(X_t), \qquad t \in [0,1]. \tag{47}$$

Proof. Conditioned on $\widetilde{X}_0$, integrating (46) gives

$$\widetilde{X}_t = h(\widetilde{X}_0, t) + \int_0^t \sqrt{\dot{\Sigma}(u)}\, dB_u. \tag{48}$$

For each fixed t, the stochastic integral is Gaussian with covariance

$$\int_0^t \dot{\Sigma}(u)\, du = \Sigma(t) - \Sigma(0) = \Sigma(t), \tag{49}$$

since $\Sigma(0) = 0$. Therefore,

$$\widetilde{X}_t = h(\widetilde{X}_0, t) + \sqrt{\Sigma(t)}\, \eta, \qquad \eta \sim \mathcal{N}(0, I). \tag{50}$$

Because $\widetilde{X}_0 \overset{d}{=} X_0$, this representation has the same marginal distribution as

$$X_t = h(X_0, t) + \sqrt{\Sigma(t)}\, \eta. \tag{51}$$

Hence $\mathcal{L}(\widetilde{X}_t) = \mathcal{L}(X_t)$ for all $t \in [0,1]$. □

A.2 Markovian projection

The process $\widetilde{X}_t$ has the desired marginal distributions but is generally non-Markovian because its drift depends on the initial variable $\widetilde{X}_0$. We therefore use Markovian projection to construct a Markov process with the same one-time marginal distributions.

Theorem A.2 (Markovian projection)**.** *Let $\widetilde{X}_t$ be the non-Markovian process in* (46)*. Assume that*

$$\mathbb{E}\int_0^1 \left(\left\| \partial_t h(\widetilde{X}_0, t) \right\| + \left\| \dot{\Sigma}(t) \right\| \right) dt < \infty. \tag{52}$$

Assume also that the projected martingale problem is well posed. Define

$$b(\mathbf{x}, t) := \mathbb{E}\Big[\partial_t h(\widetilde{X}_0, t) \mid \widetilde{X}_t = \mathbf{x} \Big]. \tag{53}$$

Consider the Markovian SDE

$$d\bar{X}_t = b(\bar{X}_t, t)\, dt + \sqrt{\dot{\Sigma}(t)}\, dB_t, \qquad \bar{X}_0 \overset{d}{=} X_0, \tag{54}$$

with $\bar{X}_0$ independent of B_t. Then $\bar{X}_t$, $\widetilde{X}_t$, and X_t have identical one-time marginal distributions for all $t \in [0,1]$:

$$\mathcal{L}(\bar{X}_t) = \mathcal{L}(\widetilde{X}_t) = \mathcal{L}(X_t), \qquad t \in [0,1]. \tag{55}$$

Proof. This is a direct application of the Markovian projection theorem of Brunick and Shreve [30]. The non-Markovian process (46) has drift $\partial_t h(\widetilde{X}_0, t)$ and diffusion coefficient $\sqrt{\dot{\Sigma}(t)}$. Markovian projection replaces the drift by its conditional expectation given the current state while preserving the diffusion coefficient. Thus the projected drift is precisely (53). The projected Markovian process has the same one-time marginal distributions as $\widetilde{X}_t$. Combining this with Theorem A.1 gives the stated equality of marginal distributions. □

A.3 Projected drift and likelihood-score approximation

We first express the projected drift in terms of conditional means.

Proposition A.3 (Projected drift representation)**.** *Define*

$$m_1(\mathbf{x}, t) := \mathbb{E}\Big[\widetilde{X}_0 \mid \widetilde{X}_t = \mathbf{x}\Big], \qquad m_2(\mathbf{x}, t) := \mathbb{E}\Big[H(\widetilde{X}_0) \mid \widetilde{X}_t = \mathbf{x}\Big]. \tag{56}$$

Then the projected drift in (53) *is*

$$b(\mathbf{x}, t) = \dot{a}(t)\,(m_1(\mathbf{x}, t) - m_2(\mathbf{x}, t)). \tag{57}$$

Proof. From the definition of h,

$$\partial_t h(\mathbf{x}, t) = \dot{a}(t)\,(\mathbf{x} - H(\mathbf{x})). \tag{58}$$

Substituting this into the projected drift gives

$$b(\mathbf{x}, t) = \mathbb{E}\Big[\dot{a}(t)\left(\widetilde{X}_0 - H(\widetilde{X}_0)\right) \mid \widetilde{X}_t = \mathbf{x}\Big] \tag{59}$$

$$= \dot{a}(t)\left(\mathbb{E}[\widetilde{X}_0 \mid \widetilde{X}_t = \mathbf{x}] - \mathbb{E}[H(\widetilde{X}_0) \mid \widetilde{X}_t = \mathbf{x}]\right), \tag{60}$$

which gives the claim. □

For nonlinear measurements, the transition from $\bar{X}_t$ to $\bar{X}_1$ is generally intractable. We approximate the endpoint transition distribution by freezing the conditional means over $[t, 1]$.

Lemma A.4 (Endpoint approximation for the likelihood score)**.** *Let $0 \le t < 1$ and suppose that the conditional means in the projected drift are frozen at $(\mathbf{x}, t)$ over the interval $[t, 1]$. Then*

$$\bar{X}_1 \mid \bar{X}_t = \mathbf{x} \approx \mathcal{N}\left(\mu_t(\mathbf{x}), \Sigma_{t\to 1}\right), \tag{61}$$

where

$$\mu_t(\mathbf{x}) := \mathbf{x} - a(t)\,(m_1(\mathbf{x}, t) - m_2(\mathbf{x}, t)), \tag{62}$$

$$\Sigma_{t\to 1} := \Sigma(1) - \Sigma(t). \tag{63}$$

The conditional covariance is approximated using only the diffusion contribution $\Sigma_{t\to 1}$.

Proof. The projected SDE is

$$d\bar{X}_u = b(\bar{X}_u, u)\,du + \sqrt{\dot{\Sigma}(u)}\,dB_u. \tag{64}$$

Using Proposition A.3 and freezing m_1, m_2 at $(\mathbf{x}, t)$ gives

$$b(\bar{X}_u, u) \approx \dot{a}(u)\,(m_1(\mathbf{x}, t) - m_2(\mathbf{x}, t)). \tag{65}$$

Thus,

$$\int_t^1 b(\bar{X}_u, u)\,du \approx (a(1) - a(t))\,(m_1(\mathbf{x}, t) - m_2(\mathbf{x}, t)) = -a(t)\,(m_1(\mathbf{x}, t) - m_2(\mathbf{x}, t)). \tag{66}$$

The diffusion accumulated over $[t, 1]$ has covariance

$$\int_t^1 \dot{\Sigma}(u)\,du = \Sigma(1) - \Sigma(t). \tag{67}$$

Combining the drift and diffusion contributions yields the endpoint approximation in (61). □

Corollary A.5 (Likelihood-score approximation for nonlinear measurements)**.** *Assume that μ_t is differentiable with respect to $\mathbf{x}$ and that $\Sigma_{t\to 1}$ is positive definite. Under the endpoint approximation in Lemma A.4, the likelihood score is approximated by*

$$g(\mathbf{z}, \mathbf{x}, t) := \nabla_{\mathbf{x}} \log \mathcal{N}\left(\mathbf{z}; \mu_t(\mathbf{x}), \Sigma_{t\to 1}\right) \tag{68}$$

$$= \left(\nabla_{\mathbf{x}} \mu_t(\mathbf{x})\right)^{\mathsf{T}} \Sigma_{t\to 1}^{-1}\left(\mathbf{z} - \mu_t(\mathbf{x})\right). \tag{69}$$

Proof. This follows by differentiating the Gaussian log-density in (61) with respect to $\mathbf{x}$ and applying the chain rule through $\mu_t(\mathbf{x})$. □

A.4 Exact likelihood score for linear measurements

We now consider the linear case, where $H(X_0) = AX_0$ for some matrix A.

Proposition A.6 (Conditional means for the linear forward process)**.** *For the linear measurement-aware forward process*

$$X_t = A(t)X_0 + \sqrt{\Sigma(t)}\,\eta, \qquad A(t) = a(t)I + \big(1 - a(t)\big)A, \tag{70}$$

the conditional means satisfy

$$m_2(\mathbf{x}, t) = A\, m_1(\mathbf{x}, t). \tag{71}$$

Proof. By linearity of H,

$$m_2(\mathbf{x}, t) = \mathbb{E}[H(X_0) \mid X_t = \mathbf{x}] = \mathbb{E}[AX_0 \mid X_t = \mathbf{x}] = A\,\mathbb{E}[X_0 \mid X_t = \mathbf{x}] = A\, m_1(\mathbf{x}, t). \tag{72}$$

□

Proposition A.7 (Score-form projected drift for linear measurements)**.** *Assume that $A(t)$ is invertible for $t \in [0, 1)$. Let p_t denote the density of X_t, and define*

$$F_t := \dot{A}(t)A(t)^{-1}. \tag{73}$$

Then the projected drift can be written as

$$b(\mathbf{x}, t) = F_t\mathbf{x} + F_t\Sigma(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x}). \tag{74}$$

Proof. For the linear Gaussian perturbation

$$X_t = A(t)X_0 + \sqrt{\Sigma(t)}\,\eta, \tag{75}$$

the denoising identity gives

$$A(t)m_1(\mathbf{x}, t) = \mathbf{x} + \Sigma(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x}). \tag{76}$$

Since $m_2 = Am_1$, Proposition A.3 yields

$$b(\mathbf{x}, t) = \dot{a}(t)(I - A)m_1(\mathbf{x}, t). \tag{77}$$

Using $\dot{A}(t) = \dot{a}(t)(I - A)$ and substituting (76), we obtain

$$b(\mathbf{x}, t) = \dot{A}(t)A(t)^{-1}\left(\mathbf{x} + \Sigma(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right), \tag{78}$$

which gives (74). □

The projected drift in (74) still depends on the prior score. To obtain an explicit endpoint transition distribution, we next construct a linear SDE with the same marginal distributions.

Theorem A.8 (Linear SDE)**.** *Assume that $A(t)$ is invertible for $t \in [0, 1)$ and let*

$$F_t = \dot{A}(t)A(t)^{-1}. \tag{79}$$

Initialize $\check{X}_0 \overset{d}{=} X_0$ independently of B_t. Consider the linear SDE

$$d\check{X}_t = F_t\check{X}_t\, dt + G_t\, dB_t, \tag{80}$$

where G_t is chosen such that

$$G_tG_t^{\mathsf{T}} = \dot{\Sigma}(t) - F_t\Sigma(t) - \Sigma(t)F_t^{\mathsf{T}}, \tag{81}$$

provided that the right-hand side is positive semidefinite, the associated Fokker–Planck equation is well posed for $t < 1$, and the transition distributions admit weak limits at $t = 1$. Transition distributions at $t = 1$ are understood through these limits. Moreover,

$$\mathcal{L}(\check{X}_t) = \mathcal{L}(X_t), \qquad t \in [0, 1]. \tag{82}$$

Proof. Since the right-hand side of (81) is positive semidefinite, there exists a matrix G_t satisfying (81). Writing $\hat{p}_t$ for the density of $\check{X}_t$, the Fokker–Planck equation of (80) is

$$\partial_t \hat{p}_t = -\nabla \cdot (F_t \mathbf{x} \hat{p}_t) + \frac{1}{2} \nabla \cdot \left(G_t G_t^{\mathsf{T}} \nabla \hat{p}_t \right). \tag{83}$$

For the known marginal density p_t, substitute $p_t \nabla \log p_t = \nabla p_t$ into the Fokker–Planck equation of the projected drift in Proposition A.7. Its diffusion terms combine to give (81), so p_t also solves the displayed linear Fokker–Planck equation. Assuming uniqueness for this equation with the common initial distribution, the marginal distributions agree. □

Theorem A.9 (Exact likelihood score for linear measurements)**.** *Assume that $A(t)$ is invertible for $t \in [0, 1)$. Under the assumptions of Theorem A.8, let Z denote the terminal variable of the linear SDE. Its transition distribution from t to 1 is*

$$Z \mid \check{X}_t \sim \mathcal{N}\left(M_{t\to 1} \check{X}_t, \check{\Sigma}_{t\to 1} \right), \tag{84}$$

where

$$M_{t\to 1} = A(1)A(t)^{-1}, \qquad \check{\Sigma}_{t\to 1} = \Sigma(1) - M_{t\to 1} \Sigma(t) M_{t\to 1}^{\mathsf{T}}. \tag{85}$$

Consequently, when $\check{\Sigma}_{t\to 1}$ is positive definite,

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{z} \mid \mathbf{x}_t) = M_{t\to 1}^{\mathsf{T}} \check{\Sigma}_{t\to 1}^{-1} \left(\mathbf{z} - M_{t\to 1} \mathbf{x}_t \right). \tag{86}$$

Proof. For $t < v < 1$, variation of constants gives

$$\check{X}_v = M_{t\to v} \check{X}_t + \int_t^v M_{u\to v} G_u \, dB_u. \tag{87}$$

Here, $M_{t\to v} = A(v)A(t)^{-1}$. The stochastic integral is independent of $\check{X}_t$ and Gaussian. Since $\partial_u M_{u\to v} = -M_{u\to v} F_u$, its covariance satisfies

$$\begin{aligned} \check{\Sigma}_{t\to v} &= \int_t^v M_{u\to v} G_u G_u^{\mathsf{T}} M_{u\to v}^{\mathsf{T}} \, du && (88) \\ &= \int_t^v \frac{d}{du} \left[M_{u\to v} \Sigma(u) M_{u\to v}^{\mathsf{T}} \right] du && (89) \\ &= \Sigma(v) - M_{t\to v} \Sigma(t) M_{t\to v}^{\mathsf{T}}. && (90) \end{aligned}$$

Taking the assumed weak endpoint limit gives (84) and (85). Differentiating the resulting Gaussian log-density gives (86). □

Remark A.10 (Invertibility of $A(t)$)*.* Let λ_i be the eigenvalues of A. Since

$$A(t) = a(t) I + \big(1 - a(t)\big) A, \tag{91}$$

the eigenvalues of $A(t)$ are

$$a(t) + \big(1 - a(t)\big) \lambda_i. \tag{92}$$

Thus, $A(t)$ is singular only if

$$a(t) = \frac{-\lambda_i}{1 - \lambda_i} \tag{93}$$

for some eigenvalue $\lambda_i \neq 1$. In particular, if all eigenvalues of A lie in $[0, 1]$, then $A(t)$ is invertible for all $t \in [0, 1)$. This condition covers common linear measurement operators such as grid masks and center masks.

A.5 Guided reverse-time sampling

Guided reverse-time sampling combines the prior score with the corresponding likelihood score based on the posterior-score decomposition. For nonlinear measurements, we use the likelihood-score approximation in (68). For linear measurements, we use the exact likelihood score derived in Section A.4.

Lemma A.11 (Guided reverse update)**.** *For nonlinear measurements and reverse-time discretization points $0 \le t < s < 1$, write $\Sigma_{s\to t} := \Sigma(t) - \Sigma(s)$. Given $\mathbf{x}_s$, define*

$$\widehat{h}_s(\mathbf{x}_s) := a(s)m_1(\mathbf{x}_s, s) + \big(1 - a(s)\big)m_2(\mathbf{x}_s, s). \tag{94}$$

Estimate the prior score by

$$s_{\text{prior}}(\mathbf{x}_s, s) := -\Sigma(s)^{-1}\left(\mathbf{x}_s - \widehat{h}_s(\mathbf{x}_s)\right). \tag{95}$$

Let

$$s_{\text{guide}}(\mathbf{x}_s, s, \mathbf{z}) := s_{\text{prior}}(\mathbf{x}_s, s) + g(\mathbf{z}, \mathbf{x}_s, s). \tag{96}$$

Using the prior and likelihood scores gives the guided reverse update

$$\mathbf{x}_t = \mathbf{x}_s + \big(a(t) - a(s)\big)\left(m_1(\mathbf{x}_s, s) - m_2(\mathbf{x}_s, s)\right) - \Sigma_{s\to t}s_{\text{guide}}(\mathbf{x}_s, s, \mathbf{z}) + \sqrt{-\Sigma_{s\to t}}\,\eta, \tag{97}$$

where $\eta \sim \mathcal{N}(0, I)$.

Proof. The first drift term follows from discretizing the projected forward drift in reverse time:

$$\int_s^t \dot{a}(u)\left(m_1(\mathbf{x}_s, s) - m_2(\mathbf{x}_s, s)\right)du \approx \big(a(t) - a(s)\big)\left(m_1(\mathbf{x}_s, s) - m_2(\mathbf{x}_s, s)\right). \tag{98}$$

The score term uses $s_{\text{guide}} = s_{\text{prior}} + g$, based on the posterior-score decomposition

$$\nabla_{\mathbf{x}_s}\log p(\mathbf{x}_s \mid \mathbf{z}) = \nabla_{\mathbf{x}_s}\log p(\mathbf{x}_s) + \nabla_{\mathbf{x}_s}\log p(\mathbf{z} \mid \mathbf{x}_s), \tag{99}$$

with the two terms approximated by s_{prior} and g, respectively. Since $\Sigma(t)$ is increasing in the positive-semidefinite order and $s > t$, $\Sigma_{s\to t} = \Sigma(t) - \Sigma(s) \preceq 0$. Thus $-\Sigma_{s\to t} \succeq 0$ defines the reverse-time covariance used both in the score correction and in the diffusion term $\sqrt{-\Sigma_{s\to t}}\,\eta$. □

Linear measurements. For the linear SDE in Theorem A.8, the implementation uses (33), with the transition parameters evaluated for $s > t$. The reverse covariance is well defined because

$$-\check{\Sigma}_{s\to t} = M_{s\to t}\check{\Sigma}_{t\to s}M_{s\to t}^{\mathsf{T}} \succeq 0. \tag{100}$$

This is a finite-step sampling rule, not an exact posterior transition.

A.6 Training conditional-mean estimators

It remains to estimate the conditional means m_1 and m_2 appearing in the projected drift and guided sampler. Although these conditional means are defined through the non-Markovian process $\widetilde{X}_t$, the explicit forward process X_t induces the same joint distribution for the pair of initial and perturbed states:

$$X_t = h(X_0, t) + \sqrt{\Sigma(t)}\,\eta, \qquad \widetilde{X}_t = h(\widetilde{X}_0, t) + \sqrt{\Sigma(t)}\,\eta, \qquad \widetilde{X}_0 \overset{d}{=} X_0. \tag{101}$$

Thus, in practice, the conditional means can be estimated using samples from the forward process in (44). The following proposition shows that the corresponding estimators are obtained by minimizing denoising L_2 objectives.

Proposition A.12 (Denoising objective for conditional means)**.** *Let*

$$X_t = h(X_0, t) + \sqrt{\Sigma(t)}\,\eta, \qquad \eta \sim \mathcal{N}(0, I). \tag{102}$$

For a fixed t, consider conditional-mean estimators $m_{1,\theta}$ and $m_{2,\theta}$ trained by

$$\mathcal{L}_{\text{reg}}(\theta, t) = \mathbb{E}_{X_0,\eta}\left[\|m_{1,\theta}(X_t, t) - X_0\|^2 + \|m_{2,\theta}(X_t, t) - H(X_0)\|^2\right]. \tag{103}$$

Minimizing over square-integrable predictors gives

$$m^\star_{1,\theta}(\mathbf{x},t) = \mathbb{E}[X_0 \mid X_t = \mathbf{x}], \qquad m^\star_{2,\theta}(\mathbf{x},t) = \mathbb{E}[H(X_0) \mid X_t = \mathbf{x}]. \tag{104}$$

These identities describe the unrestricted regression targets, which a parametric model attains only if its function class contains the conditional means. The pair (X_0, X_t) has the same distribution as $(\widetilde{X}_0, \widetilde{X}_t)$, so these targets equal the conditional means in (56)*. For the losses defined in Proposition 3.6, we also have*

$$\nabla_\theta L_i(\theta,t) = \nabla_\theta \widehat{L}_i(\theta,t), \qquad i = 1,2. \tag{105}$$

Proof. This follows from the standard L^2 projection identity. For a fixed t, the minimizer of

$$\mathbb{E}\left[\|f(X_t,t) - Y\|^2\right] \tag{106}$$

is the conditional mean $\mathbb{E}[Y \mid X_t]$. Applying this identity with $Y = X_0$ gives

$$m^\star_{1,\theta}(\mathbf{x},t) = \mathbb{E}[X_0 \mid X_t = \mathbf{x}], \tag{107}$$

and applying it with $Y = H(X_0)$ gives

$$m^\star_{2,\theta}(\mathbf{x},t) = \mathbb{E}[H(X_0) \mid X_t = \mathbf{x}]. \tag{108}$$

Because the non-Markovian process satisfies $\widetilde{X}_t = h(\widetilde{X}_0, t) + \sqrt{\Sigma(t)}\eta$ with $\widetilde{X}_0 \overset{d}{=} X_0$, the conditional distributions induced by (X_0, X_t) and $(\widetilde{X}_0, \widetilde{X}_t)$ coincide. Therefore, the denoising objectives estimate the conditional means required in the projected drift and guided sampler. For a fixed t and $i = 1, 2$, the same projection identity gives

$$\widehat{L}_i(\theta,t) = L_i(\theta,t) + \mathbb{E}\Big[\|y_i(X_0) - m_i(X_t,t)\|^2\Big]. \tag{109}$$

The cross term vanishes because $m_i(X_t,t) = \mathbb{E}[y_i(X_0) \mid X_t]$. The final term is independent of θ, which yields the stated gradient equality. □

For linear measurements, $m_2 = Am_1$ by (71), so we minimize only $\widehat{L}_1(\theta,t)$ and set $m_{2,\theta} = Am_{1,\theta}$.

Measurement targets. For nonlinear measurements, the target is $H(X_0)$. Let $Y_2 = H(X_0) + \varepsilon$, where $\mathbb{E}[\varepsilon \mid X_0, X_t] = 0$ and $\mathbb{E}\|\varepsilon\|^2 < \infty$. Then

$$\mathbb{E}[Y_2 \mid X_t] = \mathbb{E}[H(X_0) \mid X_t] = m_2(X_t,t), \tag{110}$$

$$\mathbb{E}\|m_{2,\theta}(X_t,t) - Y_2\|^2 = \mathbb{E}\|m_{2,\theta}(X_t,t) - H(X_0)\|^2 + \mathbb{E}\|\varepsilon\|^2. \tag{111}$$

Thus replacing $H(X_0)$ by Y_2 changes the loss only by a parameter-independent constant and preserves the conditional-mean target. The target $H(X_0)$ is computed from prior samples and is distinct from the observed measurement being assimilated.

B Algorithmic details

This section describes the practical data assimilation procedure used by MASF. We describe dataset construction, pretraining, online training, measurement updates, and time updates.

B.1 MASF procedure

Dataset construction. Let $\{\tau_k\}_{k=1}^K$ denote the measurement time grid. For filtering, we sample an initial ensemble from the prior distribution,

$$\{\hat{\mathbf{x}}^{(i)}_{\text{init}}\}_{i=1}^N \sim p_{\texttt{init}}, \tag{112}$$

and propagate each ensemble member to the first measurement time:

$$\hat{\mathbf{x}}^{(i)}_1 = \Phi_{\tau_1}\left(\hat{\mathbf{x}}^{(i)}_{\text{init}}\right), \qquad i = 1,\dots,N. \tag{113}$$

Here, Φ_{τ_1} denotes the numerical flow map to τ_1, implemented by repeated applications of the dataset-specific transition solver. For stochastic state equations, each simulated trajectory uses independent process-noise draws.

For evaluation, we independently sample a ground-truth initial state

$$\mathbf{x}_{\text{init}}^{\text{gt}} \sim p_{\texttt{init}} \tag{114}$$

and propagate it along the measurement time grid:

$$\mathbf{x}_k^{\text{gt}} = \Phi_{\tau_k}\left(\mathbf{x}_{\text{init}}^{\text{gt}}\right), \qquad k = 1, \ldots, K. \tag{115}$$

The measurement at time τ_k is generated as

$$\mathbf{z}_k = H(\mathbf{x}_k^{\text{gt}}) + \sqrt{R}\eta_k, \qquad \eta_k \sim \mathcal{N}(0, I), \qquad k = 1, \ldots, K. \tag{116}$$

Given the state equation and measurement operator, MASF uses the measurements and its own ensemble during filtering. The independent ground-truth trajectory is used only for evaluation.

Pretraining. Before filtering, we pretrain the model on samples generated from the state equation, without online measurements.

Online training at a measurement time. At each measurement time τ_k, we fine-tune the conditional-mean estimators using the prior ensemble $\{\hat{\mathbf{x}}_k^{(i)}\}_{i=1}^N$, starting from pretrained weights for $k = 1$ and previous-step weights thereafter. For this fine-tuning, we perturb the ensemble through the measurement-aware forward process:

$$\hat{\mathbf{x}}_{k,t}^{(i)} = h(\hat{\mathbf{x}}_k^{(i)}, t) + \sqrt{\Sigma(t)}\,\eta^{(i)}, \qquad \eta^{(i)} \sim \mathcal{N}(0, I). \tag{117}$$

For nonlinear measurements, we train $m_{1,\theta}$ and $m_{2,\theta}$ using (103). For linear measurements $H(\mathbf{x}) = A\mathbf{x}$, we train only $m_{1,\theta}$, since $m_{2,\theta} = Am_{1,\theta}$ is available analytically.

Measurement update. After training at τ_k, MASF performs the measurement update using guided reverse-time sampling. Starting from the prior ensemble $\{\hat{\mathbf{x}}_k^{(i)}\}_{i=1}^N$, we initialize the reverse sampler at s_0 using the measurement-aware forward process, following the sampling convention in Appendix C.4:

$$\mathbf{x}_{s_0}^{(i)} = h(\hat{\mathbf{x}}_k^{(i)}, s_0) + \sqrt{\Sigma(s_0)}\,\eta^{(i)}, \qquad \eta^{(i)} \overset{\text{iid}}{\sim} \mathcal{N}(0, I). \tag{118}$$

Given the measurement $\mathbf{z}_k$, the sampler produces an approximate posterior ensemble:

$$\mathbf{x}_k^{(i)} = \texttt{GuidedSampler}\left(\mathbf{x}_{s_0}^{(i)}, \mathbf{z}_k; \theta_k\right), \qquad i = 1, \ldots, N. \tag{119}$$

The sampler combines the prior score with the likelihood score induced by $\mathbf{z}_k$. For nonlinear measurements, the likelihood score is computed using (68). For linear measurements, we use the exact likelihood score in (86). The number of function evaluations (NFE) determines the number of reverse-time updates.

Time update. After assimilating the measurement at τ_k, MASF propagates the posterior ensemble to the next measurement time. Let n_k denote the number of numerical transition steps between τ_k and τ_{k+1}. The next prior ensemble is

$$\hat{\mathbf{x}}_{k+1}^{(i)} = \Phi_{n_k}\left(\mathbf{x}_k^{(i)}\right), \qquad i = 1, \ldots, N, \tag{120}$$

where Φ_{n_k} denotes n_k repeated applications of the transition solver.

The full procedure is summarized in Algorithm 1. The covariance floor and pseudo-inverse option are described in Appendix C.4.

Algorithm 1 Measurement-Aware Score-Based Filter

Input: measurements $(\mathbf{z}_k)_{k=1}^K$, ensemble size N, epochs E, NFE
Input: schedule $a(t)$, covariance $\Sigma(t)$, measurement operator H, transition solver Φ
Input: offline-pretrained weights
Input: training interval $[0, T]$, sampling start s_0, terminal time $t_{\min}$
Output: ensemble-mean estimates $(\bar{\mathbf{x}}_k)_{k=1}^K$
Sample $(\hat{\mathbf{x}}_{\text{init}}^{(i)})_{i=1}^N \sim p_{\texttt{init}}$
$\hat{\mathbf{x}}_1^{(i)} \leftarrow \Phi_{\tau_1}(\hat{\mathbf{x}}_{\text{init}}^{(i)}), \quad i = 1, \dots, N$
for $k = 1$ **to** K **do**
  if $k = 1$ **then**
    Initialize θ_1 from the offline-pretrained weights
  else
    $\theta_k \leftarrow \theta_{k-1}$
  end if
  for $e = 1$ **to** E **do**
    Sample $t \sim \mathcal{U}(0, T)$ and $\eta^{(i)} \sim \mathcal{N}(0, I)$
    $\hat{\mathbf{x}}_{k,t}^{(i)} \leftarrow h(\hat{\mathbf{x}}_k^{(i)}, t) + \sqrt{\Sigma(t)}\, \eta^{(i)}$
    if H is linear **then**
      $L \leftarrow \frac{1}{N} \sum_{i=1}^N \| m_{1,\theta_k}(\hat{\mathbf{x}}_{k,t}^{(i)}, t) - \hat{\mathbf{x}}_k^{(i)} \|^2$
    else
      Set $y_2^{(i)} \leftarrow H(\hat{\mathbf{x}}_k^{(i)})$
      $L \leftarrow \frac{1}{N} \sum_{i=1}^N \big[\| m_{1,\theta_k}(\hat{\mathbf{x}}_{k,t}^{(i)}, t) - \hat{\mathbf{x}}_k^{(i)} \|^2 + \| m_{2,\theta_k}(\hat{\mathbf{x}}_{k,t}^{(i)}, t) - y_2^{(i)} \|^2 \big]$
    end if
    Update θ_k by minimizing L
  end for
  Set s_0 according to Appendix C.4
  Initialize the reverse-time ensemble using (118)
  $(\mathbf{x}_k^{(i)})_{i=1}^N \leftarrow \texttt{GuidedSampler}((\mathbf{x}_{s_0}^{(i)})_{i=1}^N, \mathbf{z}_k, \theta_k, H, a, \Sigma, \text{NFE}, t_{\min})$
  $\bar{\mathbf{x}}_k \leftarrow \frac{1}{N} \sum_{i=1}^N \mathbf{x}_k^{(i)}$
  if $k < K$ **then**
    Let n_k be the number of transition steps from τ_k to τ_{k+1}
    $\hat{\mathbf{x}}_{k+1}^{(i)} \leftarrow \Phi_{n_k}(\mathbf{x}_k^{(i)}), \quad i = 1, \dots, N$
  end if
end for

Algorithm 2 `GuidedSampler`

Input: ensemble $(\mathbf{x}_{s_0}^{(i)})_{i=1}^N$, measurement $\mathbf{z}_k$, parameters θ_k, measurement operator H, schedules a, Σ, NFE
Input: terminal time $t_{\min}$
Output: approximate posterior ensemble $(\mathbf{x}_k^{(i)})_{i=1}^N$
Construct a reverse-time grid $s_0 > s_1 > \cdots > s_L = t_{\min}$ with $L = \text{NFE}$
for $\ell = 0$ **to** $L - 1$ **do**
  Set $s \leftarrow s_\ell$ and $t \leftarrow s_{\ell+1}$
  Evaluate $m_{1,\theta_k}(\mathbf{x}_s^{(i)}, s)$
  if H is linear **then**
    Set $\widehat{h}_s = A(s) m_{1,\theta_k}(\mathbf{x}_s^{(i)}, s)$ and compute the prior score using (30)
    Compute g using (28)
  else
    Evaluate $m_{2,\theta_k}(\mathbf{x}_s^{(i)}, s)$ and compute the prior score using (30)
    Compute g using (23)
  end if
    Form $s_{\text{guide}}(\mathbf{x}_s^{(i)}, s, \mathbf{z}_k)$ using (31)
  if H is linear **then**
    Update $\mathbf{x}_t^{(i)}$ using (33)
  else
    Update $\mathbf{x}_t^{(i)}$ using (32)
  end if
  end for
  return $\mathbf{x}_k^{(i)} \leftarrow \mathbf{x}_{s_L}^{(i)}, \quad i = 1, \dots, N$

C Implementation and tuning details

This section specifies evaluation metrics, benchmark settings, implementation, and hyperparameter tuning.

C.1 Evaluation metrics

Metrics are computed at measurement times for Lorenz96 and at every step for Kolmogorov flow. Let $n = 1, \dots, T$ index these evaluation times. At each evaluation time, the state estimate is the ensemble mean:

$$\bar{\mathbf{x}}_n = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_n^{(i)}. \tag{121}$$

Given the ground-truth trajectory $\{\mathbf{x}_n^{\mathrm{gt}}\}_{n=1}^{T}$, we report the root mean squared error (RMSE) computed over these evaluation times:

$$\mathrm{RMSE} = \sqrt{\frac{1}{Td} \sum_{n=1}^{T} \left\| \bar{\mathbf{x}}_n - \mathbf{x}_n^{\mathrm{gt}} \right\|_2^2}. \tag{122}$$

For an ensemble of $N > 1$ members $x_{n,j}^{(i)}$ and truth $x_{n,j}^{\mathrm{gt}}$, the fair CRPS [36] at time n and coordinate j is

$$\mathrm{fCRPS}_{n,j} = \frac{1}{N} \sum_{i=1}^{N} \left| x_{n,j}^{(i)} - x_{n,j}^{\mathrm{gt}} \right| - \frac{1}{N(N-1)} \sum_{i=1}^{N} (2i - N - 1) x_{n,j}^{(i:N)}, \tag{123}$$

where $x_{n,j}^{(i:N)}$ is the ith smallest ensemble member at coordinate (n, j). This sorted form equals the pairwise-distance fair CRPS. The reported fCRPS averages these marginal scores over state coordinates and evaluation times. The denominator $N(N-1)$ corrects the pairwise term for finite ensemble size, making comparison across different ensemble sizes more appropriate than the empirical-CRPS denominator N^2. Define the unbiased ensemble variance and aggregate spread by

$$s_{n,j}^2 = \frac{1}{N-1} \sum_{i=1}^{N} \left(x_{n,j}^{(i)} - \bar{x}_{n,j} \right)^2, \tag{124}$$

$$\mathrm{spread} = \sqrt{\frac{1}{Td} \sum_{n=1}^{T} \sum_{j=1}^{d} s_{n,j}^2}. \tag{125}$$

The spread–skill ratio includes the finite-ensemble correction [37]

$$\mathrm{SSR} = \sqrt{\frac{N+1}{N}} \, \frac{\mathrm{spread}}{\mathrm{RMSE}}. \tag{126}$$

Spread and RMSE are aggregated over all evaluation times and state coordinates before taking their ratio; we do not average time-wise ratios. Values below one indicate insufficient spread, values above one indicate excessive spread, and values near one indicate agreement between spread and error. Neither fCRPS nor SSR alone verifies the joint dependence structure of the posterior.

C.2 Benchmark settings

C.2.1 Lorenz96

The Lorenz96 experiment uses $d = 40$, forcing $F = 8$, and periodic indices in (37). We integrate the drift with fourth-order Runge–Kutta (RK4) and step size $\Delta\tau = 0.05$. At each integration step, independent process noise is added according to

$$\mathbf{x}_{n+1} = \Psi_{\Delta\tau}^{\mathrm{RK4}}(\mathbf{x}_n) + \sigma_p \sqrt{\Delta\tau}\, \boldsymbol{\xi}_n, \qquad \boldsymbol{\xi}_n \sim \mathcal{N}(0, I_{40}), \tag{127}$$

where $\Psi_{\Delta\tau}^{\mathrm{RK4}}$ denotes one RK4 step for the drift and $\sigma_p = 0.1$. The initial prior is $\mathbf{x}_0 \sim \mathcal{N}(0, I_{40})$. The first 25 integration steps ($\tau = 1.25$) form the spin-up period. We assimilate 11 measurements at steps $25, 35, \ldots, 125$, with a measurement interval of 0.5 time units. All 40 components are measured with the monotonic cubic operator in (38) and additive physical-space noise:

$$\mathbf{z}_k = H(\mathbf{x}_k) + \varepsilon_k, \qquad \varepsilon_k \sim \mathcal{N}(0, 0.1^2 I_{40}). \tag{128}$$

All methods use $N = 100$ ensemble members. For MASF, we use a dual-head one-dimensional U-Net with model dimension 64, initial dimension 32, and channel multipliers $(1, 2, 4)$. We pretrain this model with batch size 32 for 200 epochs using a learning rate of 3×10^{-4}. Online fine-tuning uses 100 full epochs and a learning rate of 5×10^{-5}, and reverse-time sampling uses $\mathrm{NFE} = 200$. For Lorenz96, the guided reverse-time update uses the ratio $a(t)/a(s)$, with $m_{2,\theta}$ evaluated at $(\mathbf{x}_s, s)$. MASF uses ensemble inflation only for Lorenz96. Immediately after posterior sampling at each measurement time, we rescale the ensemble deviations from their mean by 1.5, using $\mathbf{x}_k^{(i)} \leftarrow \bar{\mathbf{x}}_k + 1.5\big(\mathbf{x}_k^{(i)} - \bar{\mathbf{x}}_k\big)$. The inflated ensemble is then propagated under the state equation to form the next prior ensemble, with no additional inflation applied to the prior ensemble or training data. Table 1 reports means and standard deviations across 20 seeds (42–61), using the SSR estimator in (126).

C.2.2 Kolmogorov flow

We simulate the Kolmogorov flow on $[0, 2\pi]^2$ using JAX-CFD [33], with density $\rho = 1$, Reynolds number $\mathrm{Re} = 1000$, and periodic boundary conditions. The external forcing is $\mathbf{f} = (\sin(4y), 0)^{\mathsf{T}} - 0.1\mathbf{u}$, where y is the second spatial coordinate and the second term provides linear damping. Successive states are separated by $\Delta\tau = 0.2$, with each interval integrated by the semi-implicit Navier–Stokes solver using smaller time steps chosen for stability. Unless otherwise specified, measurements are taken every 10 steps over steps 50–100, with additive measurement noise with standard deviation $\sigma = 0.1$. The grid and center masks are defined by

$$\begin{aligned} [A_{\mathrm{grid}}]_{ij} &= \mathbf{1}\{i \equiv 0 \pmod{s},\ j \equiv 0 \pmod{s}\}, \\ [A_{\mathrm{center}}]_{ij} &= \mathbf{1}\{(i,j) \notin \mathcal{C}_h\}, \end{aligned} \tag{129}$$

where s is the grid-mask stride and $\mathcal{C}_h$ is the central square hole of side length h. Both masks are applied element-wise to the state; their state-space embedding and noise treatment are described in Appendix C.3. The speed operator in Section 4.2 applies the sigmoid to the squared local speed. For these experiments, MASF uses $N = 10$ ensemble members and $\mathrm{NFE} = 150$–300, with one-time pretraining on 1000 samples generated from the state equation. Architecture and training details are provided in Appendices C.5 and C.7.

C.3 Measurement representation

For grid and center masks, let $P \in \mathbb{R}^{m \times d}$ select the observed coordinates. The clean measurement is embedded as

$$H(\mathbf{x}) = P^{\mathsf{T}} P \mathbf{x}, \tag{130}$$

which retains observed coordinates and sets the remaining coordinates to zero. After embedding, we add independent noise at every state-space coordinate, including the unobserved coordinates. Equivalently, an observed measurement $Z_{\mathrm{obs}} = PX + \sigma\eta_{\mathrm{obs}}$ with $\eta_{\mathrm{obs}} \sim \mathcal{N}(0, I_m)$ is augmented by independent auxiliary noise as

$$Z = P^{\mathsf{T}} Z_{\mathrm{obs}} + \sigma(I_d - P^{\mathsf{T}} P)\eta_{\mathrm{aux}}, \qquad \eta_{\mathrm{aux}} \sim \mathcal{N}(0, I_d). \tag{131}$$

The auxiliary noise is independent of the state and observed measurements, and the resulting physical-space covariance is $R = \sigma^2 I_d$. Because $Z_{\mathrm{obs}} = PZ$ and the auxiliary noise is independent, conditioning on Z gives the same state posterior as conditioning on Z_{obs}. For these binary masks, the direct likelihood score is zero on unobserved coordinates. Channel-wise normalization transforms the measurement operator and noise covariance consistently; the implementation uses this transformed covariance in guidance.

Speed measurements. The scalar observable in (41) uses the shift parameter $\kappa = 3$. The clean scalar field is replicated into two channels before measurement noise is added:

$$Z_{cij} = [H(\mathbf{x})]_{cij} + \varepsilon_{cij}, \qquad \varepsilon_{cij} \overset{\mathrm{iid}}{\sim} \mathcal{N}(0, 0.1^2), \quad c = 1, 2. \tag{132}$$

The clean field contains HW distinct scalar values, while the noisy measurement has shape $2\times H\times W$ because the two copies receive independent noise. This operator loses velocity-direction information even though its stored output has the same shape as the state.

C.4 Noise schedule and covariance regularization

We use the cosine schedule [38]

$$a(t)=\frac{\cos\left(\frac{t+s_{\cos}}{1+s_{\cos}}\frac{\pi}{2}\right)}{\cos\left(\frac{s_{\cos}}{1+s_{\cos}}\frac{\pi}{2}\right)}, \tag{133}$$

with $s_{\cos}=0.008$. The forward process reaches the measurement space at $t=1$. Training times are drawn from $[0,T]$ with an experiment-specific upper bound T. Reverse-time sampling starts slightly before $t=1$ and terminates at $t_{\min}$.

For covariance inversion, we add a small diagonal floor and use

$$\Sigma^{-1}\leftarrow(\Sigma+\delta I)^{-1}, \tag{134}$$

with $\delta=10^{-5}$. We do not use a pseudo-inverse unless otherwise specified.

C.5 Efficient implementation

For the Kolmogorov-flow experiments, we reduce the cost of online model training and reverse-time sampling through three implementation strategies: a lightweight model, amortized pretraining, and fewer function evaluations.

Lightweight architecture. During online fine-tuning, the model uses the current prior ensemble to estimate the conditional means required by the projected drift and guided sampler. Thus, the learning problem is localized around the prior distribution at the current measurement time. Unlike unconditional score-based generative modeling, this localized task does not require learning a global data distribution for open-ended generation. This allows us to use a lightweight U-Net rather than a large diffusion backbone. For linear measurements, we further reduce the output dimension by training only $m_{1,\theta}$, since $m_2=Am_1$ is available analytically. For nonlinear measurements, we use a dual-head U-Net with a shared encoder and bottleneck and separate decoders for $m_{1,\theta}$ and $m_{2,\theta}$.

Amortized pretraining. Training a conditional-mean model from scratch at every measurement time is expensive. To reduce this cost, we pretrain the conditional-mean model before filtering using samples generated from the state equation, without using online measurements. For the Kolmogorov-flow experiments, we draw 1000 initial states from $p_{\texttt{init}}$ and propagate them under the state equation to the first measurement time τ_1. The resulting prior samples form the offline pretraining dataset. The pretrained weights initialize the model at the first measurement time, avoiding training from scratch. We then retain the updated weights and fine-tune them on the current prior ensemble at each subsequent measurement time. Because consecutive measurement times have related prior distributions, this amortized initialization substantially reduces the number of online optimization steps needed for stable conditional-mean estimation. The offline pretraining cost is incurred once per configuration and is excluded from the reported online wall-clock time.

Reduced NFE. Score-based filtering methods can be expensive at inference time because reverse-time sampling typically requires many function evaluations. MASF reduces this cost by using measurement-aware likelihood guidance, which directly steers ensemble members toward the measurement-conditioned posterior. In the tested settings, MASF uses fewer function evaluations than SF and SSLS. In our experiments, MASF uses $\mathrm{NFE}=150$–300 depending on the resolution and measurement setting, whereas SF and SSLS are run with $\mathrm{NFE}=500$. The selected NFE values are chosen through the sequential tuning procedure in Appendix C.6. This reduction directly lowers sampling wall-clock time, while maintaining filtering accuracy in the reported experiments.

C.6 MASF sequential tuning procedure

We tuned MASF for Kolmogorov flow sequentially rather than jointly over all hyperparameters. The tuning procedure was automated in code using Optuna [39], with each phase run for a fixed

number of trials specified by the corresponding `n_trials` setting. For each trial, Optuna sampled a candidate configuration from the predefined search space, and completed trials were ranked by the final filtering metric. After each phase, the best-performing configuration was used as the base configuration for the next phase. The tuning phases were normalization selection, model selection, pretraining, sampling, and online fine-tuning. Unless otherwise stated, each phase used $N = 10$ ensemble members, pretrained initialization, and the assimilation window from step 50 to step 100 with a measurement interval of 10 simulation steps. Thus, configurations were selected based on final filtering performance rather than training loss alone.

Normalization selection. We first tune the normalization module because the measurement-aware forward process interpolates between state and measurement variables. If these variables have different scales or statistics, the forward process can become poorly conditioned and the conditional-mean estimators may become unstable. Given data $x \in \mathbb{R}^{B\times C\times\cdots}$, where B is the batch size and C is the number of channels, normalization statistics are computed per channel over all non-channel dimensions.

Parameter	Search values
same normalization	{`true`, `false`}
normalization form	{`affine`, `scale_only`}
statistics mode	{`standard`, `robust`, `minmax`}
statistics update	{`moving`, `fixed`, `adaptive`}
momentum	{0.2, 0.4}

Table 5: **Normalization search space for MASF.**

The parameter *same normalization* determines whether the measurement variable shares normalization statistics with the state variable. When it is set to `true`, both X and Z are normalized using statistics computed from the state ensemble. When it is set to `false`, separate measurement statistics are computed after applying the measurement operator to the state ensemble. The *normalization form* determines whether variables are centered and scaled or only rescaled:

$$x_{\text{affine}} = \frac{x-u}{s+\epsilon}, \qquad x_{\text{scale}} = \frac{x}{s+\epsilon}, \tag{135}$$

where u and s are the per-channel center and scale, and $\epsilon > 0$ stabilizes division. The *statistics mode* determines how u and s are computed. The `standard` mode uses the mean and standard deviation, the `robust` mode uses the median and median absolute deviation, and the `minmax` mode uses the midpoint and half-range of the channel values. The *statistics update* determines how normalization statistics evolve across measurement times. The `fixed` mode keeps the initial statistics fixed, the `adaptive` mode recomputes them at each update, and the `moving` mode uses an exponential moving average:

$$\theta_{\text{new}} = (1-\rho)\theta_{\text{old}} + \rho\theta_{\text{current}}, \tag{136}$$

where θ represents either normalization statistic and ρ is the momentum.

Model selection. After fixing the normalization scheme, we tune the U-Net architecture. We search over width, depth, channel-multiplier schedule, and `attention_resolutions`. Because the computational budget and spatial resolution differ across the 128^2, 256^2, and 512^2 resolutions, we use resolution-specific search spaces.

The *base channels* parameter controls the width of the U-Net. The *residual blocks* parameter determines the number of residual blocks at each resolution. The *channel multiplier* specifies how the number of channels changes across resolutions. The `attention_resolutions` specifies the spatial downsampling factor at which self-attention is added. Only factors reached by the downsampling path add attention at a U-Net level. Attention is also applied in the middle block. The selected architectures are measurement- and resolution-dependent and are reported in Tables 10 and 11.

Pretraining. We next tune pretraining hyperparameters while fixing the number of pretraining samples to 1000. The batch size, number of epochs, and learning rate control the optimization

Parameter	Kolmogorov-128	Kolmogorov-256	Kolmogorov-512
base channels	{16, 32}	{16, 32}	{16, 32}
residual blocks	{1, 2, 3}	{1, 2, 3}	{1, 2, 3}
channel multiplier	{1-2-4, 1-2-4-8}	{1-2-4-8, 1-1-2-3-4}	{1-2-4-8, 1-1-2-3-4}
attention_resolutions	{8, 16}	{8, 16, 32}	{8, 16, 32}

Table 6: **Model architecture search spaces for MASF on Kolmogorov datasets.**

budget. The validation ratio is the fraction of samples held out for validation. For pretraining, we use Adam with weight decay 0.01 and betas $(0.9, 0.99)$. The selected pretraining configurations are measurement- and resolution-dependent and are reported in Tables 10 and 11.

Parameter	Search values
pretraining samples	{1000}
batch size	{16, 32, 64}
epochs	{100, 200, 300, 400}
learning rate	{0.0001, 0.0003, 0.0005}

Table 7: **Pretraining search space for MASF.**

Sampling. We tune the sampling configuration to reduce wall-clock time while preserving filtering accuracy.

Parameter	Search values
NFE	{150, 200, 250, 300}

Table 8: **Sampling search space for MASF.**

The *NFE* controls the number of reverse-time discretization steps.

Online fine-tuning. Finally, we tune the online fine-tuning parameters used during assimilation. The model is initialized from pretrained weights at the first measurement time and fine-tuned from the previous-step weights thereafter. The search space is summarized in Table 9.

Parameter	Search values
batch size	{10}
learning rate	$\{3 \times 10^{-4}, 1 \times 10^{-4}, 5 \times 10^{-5}\}$
full epochs	{80, 90, 100, 110, 120, 150}

Table 9: **Online fine-tuning search space for MASF.**

C.7 Final MASF configuration

The selected MASF hyperparameters for the main Kolmogorov experiments are summarized in Tables 10 and 11. Table 10 reports the grid-mask configurations across resolutions, and Table 11 reports the remaining 128^2 measurement settings. We report the parameters that define the selected experimental configurations; omitted parameters are kept at their default values.

Category	Field	Grid mask 128^2	Grid mask 256^2	Grid mask 512^2
measurement	type	`grid_mask`	`grid_mask`	`grid_mask`
	same normalization	`true`	`false`	`false`
	normalization form	`affine`	`affine`	`scale_only`
	statistics mode	`standard`	`standard`	`standard`
	statistics update	`adaptive`	`moving`	`adaptive`
	main parameter	stride $= 10$	stride $= 15$	stride $= 20$
	momentum	–	0.2	–
model	type	U-Net	U-Net	U-Net
	input channels	2	2	2
	base channels	32	16	16
	residual blocks	2	1	1
	`attention_resolutions`	8	32	32
	channel multiplier	$(1, 2, 4, 8)$	$(1, 2, 4, 8)$	$(1, 2, 4, 8)$
pretraining	batch size	16	16	16
	epochs	400	300	300
	learning rate	0.0003	0.0005	0.0005
	validation ratio	0.0	0.1	0.1
sampling	NFE	200	200	150
fine-tuning	batch size	10	10	10
	learning rate	0.00005	0.0003	0.0003
	full epochs	150	110	110
	validation ratio	0.0	0.1	0.1
	weight decay	0.01	0.01	0.01

Table 10: **Selected MASF configurations for grid-mask measurements across resolutions.**

C.8 Baseline tuning ranges

For fair comparison, we tuned the filtering baselines on Kolmogorov flow over their main filtering hyperparameters. For EnKF and LETKF, we searched over ensemble size, inflation, and numerical stabilization parameters. For LETKF, we additionally tuned the localization radius.

EnKF. For EnKF, we used the stochastic full-covariance update. The search space is summarized in Table 12. The inflation factor rescales the prior covariance to mitigate ensemble underdispersion. The parameter ϵ is used for numerical stabilization in covariance operations.

LETKF. For LETKF, we tuned the ensemble size, inflation factor, localization radius, and numerical stabilization parameter. The search space is summarized in Table 13. The localization radius controls the spatial support of each local analysis update. The inflation factor rescales the prior covariance, and ϵ is used for numerical stabilization.

To choose default ensemble sizes for the main experiments, we also evaluated the accuracy–cost trade-off across ensemble sizes under grid-mask measurements. Table 14 summarizes the results. For EnKF, increasing the ensemble size improves RMSE, but larger ensembles substantially increase wall-clock time. We therefore use $N = 250$ as the default EnKF setting, which provides strong accuracy before the cost of larger ensembles becomes high. For LETKF, the RMSE improvement beyond $N = 40$ is modest compared with the additional wall-clock time, so we use $N = 40$ as the default LETKF setting.

SF and SSLS. SF and SSLS used the same backbone as MASF whenever applicable. Both SF and SSLS were run with $\mathrm{NFE} = 500$. For SSLS, we tuned the perturbation standard deviation, Langevin stepsize, and tolerance parameter. The SF and SSLS settings are summarized in Table 15.

Category	Field	Center mask	Speed
measurement	type	`center_mask`	`speed`
	same normalization	`false`	`false`
	normalization form	`scale_only`	`scale_only`
	statistics mode	`standard`	`robust`
	statistics update	`fixed`	`adaptive`
	main parameter	hole ratio $= 0.5$	$-$
model	type	U-Net	dual-head U-Net
	input channels	2	2
	base channels	32	32
	residual blocks	2	2
	`attention_resolutions`	$(16, 8)$	16
	channel multiplier	$(1, 2, 4, 8)$	$(1, 2, 4, 8)$
pretraining	batch size	16	32
	epochs	400	200
	learning rate	0.0003	0.0003
	validation ratio	0.1	0.1
sampling	NFE	300	200
fine-tuning	batch size	10	10
	learning rate	0.00005	0.0003
	full epochs	150	120
	validation ratio	0.1	0.0
	weight decay	0.01	0.0

Table 11: **Selected MASF configurations for the remaining 128^2 measurement settings.**

Parameter	Search values
ensemble size	{100, 200, 250, 300, 400}
update mode	{`full`}
inflation	{1.0, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3}
epsilon	$\{10^{-5}, 10^{-6}\}$

Table 12: **EnKF hyperparameter search space.**

Parameter	Search values
ensemble size	{10, 20, 30, 40, 60, 100}
inflation	{1.0, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3}
localization radius	{3, 5, 7}
epsilon	$\{10^{-5}, 10^{-6}\}$

Table 13: **LETKF hyperparameter search space.**

Method	Ensemble size	RMSE (↓)	Wall-clock time (s)
EnKF	40	1.01 ± 0.07	42.9 ± 2.2
	100	0.54 ± 0.19	70.6 ± 1.6
	200	0.30 ± 0.13	139.1 ± 0.6
	250	$\mathbf{0.18 \pm 0.04}$	$\mathbf{180.3 \pm 0.4}$
	400	0.14 ± 0.03	263.5 ± 9.8
LETKF	10	0.22 ± 0.02	273.1 ± 4.0
	20	0.18 ± 0.01	294.7 ± 14.9
	30	0.18 ± 0.01	308.1 ± 13.5
	40	$\mathbf{0.17 \pm 0.01}$	$\mathbf{389.6 \pm 5.4}$
	60	0.17 ± 0.02	455.9 ± 14.0

Table 14: **Ensemble-size sensitivity for EnKF and LETKF under grid-mask measurements on Kolmogorov flow at 128^2 resolution.** Mean $\pm$ standard deviation over five seeds. The selected default ensemble sizes are highlighted in bold. Time denotes online wall-clock time.

Method	Parameter	Search values / selected value
SF, SSLS	NFE	500
SSLS	perturbation std.	{0.001, 0.05, 0.1, 0.2}
SSLS	stepsize	{0.0001, 0.001, 0.002, 0.003, 0.004}
SF, SSLS	tolerance	{10, 50, 100, 200}

Table 15: **Sampling settings for SF and SSLS.**

For SSLS, the perturbation standard deviation controls the artificial perturbation scale. The stepsize controls the Langevin update size, and the tolerance parameter clips or stabilizes large guided-score updates during sampling.

D Additional experimental results

D.1 Comparison with Score-based Data Assimilation

Score-based Data Assimilation (SDA) [19] addresses joint trajectory inference rather than sequential filtering, so it is excluded from the main benchmark. For SDA, we adapt its model architecture and hyperparameters to our dataset. Table 16 compares SDA with MASF. MASF achieves lower RMSE across the three measurement settings. MASF RMSE values follow Table 2.

Measurement	SDA RMSE (↓)	MASF RMSE (↓)
Grid mask	1.06 ± 0.06	0.15 ± 0.02
Center mask	0.30 ± 0.01	0.18 ± 0.04
Speed	2.61 ± 0.32	0.27 ± 0.04

Table 16: **Comparison with SDA on Kolmogorov flow at 128^2 resolution.** RMSE is compared under grid-mask, center-mask, and speed measurements. Mean $\pm$ standard deviation over five seeds.

D.2 Evaluation under Reynolds-number shifts

We evaluate Kolmogorov-flow filtering at 128×128 resolution with grid-mask measurements over five random seeds, using ground-truth trajectories and measurements generated at $\mathrm{Re} \in \{800, 1200, 1500\}$. The MASF model is pretrained only once at $\mathrm{Re} = 1000$, and the same pretrained checkpoint is used for all shifted evaluations. The state equation remains fixed at $\mathrm{Re} = 1000$ throughout filtering. MASF adapts through its original online fine-tuning procedure; no separate offline pretraining or hyperparameter retuning is performed using shifted-regime data. Table 17 reports the mean and worst-case results over the three shifted Reynolds numbers. MASF achieves the lowest mean and worst-case RMSE and fCRPS among the evaluated methods, indicating better accuracy and marginal predictive quality under the tested shifts.

Method	N	Mean RMSE	Worst RMSE	Mean fCRPS	Worst fCRPS
EnKF	250	0.168 ± 0.014	0.206 ± 0.037	0.086 ± 0.009	0.112 ± 0.025
LETKF	40	0.166 ± 0.007	0.180 ± 0.010	0.083 ± 0.003	0.089 ± 0.004
SF	100	0.810 ± 0.023	0.835 ± 0.018	0.455 ± 0.013	0.470 ± 0.013
SSLS	100	0.504 ± 0.018	0.534 ± 0.009	0.268 ± 0.011	0.286 ± 0.007
MASF	10	$\mathbf{0.146 \pm 0.012}$	$\mathbf{0.153 \pm 0.012}$	$\mathbf{0.069 \pm 0.005}$	$\mathbf{0.073 \pm 0.005}$

Table 17: **Evaluation under Reynolds-number shifts with grid-mask measurements on Kolmogorov flow at 128^2 resolution.** Mean $\pm$ standard deviation over five seeds. Mean and worst-case results are aggregated over $\mathrm{Re} \in \{800, 1200, 1500\}$.

D.3 Qualitative results

Figures 4–6 show qualitative comparisons for grid-mask, center-mask, and speed measurements. Each figure presents the ground truth, measurements, and ensemble means from each method at selected time steps.

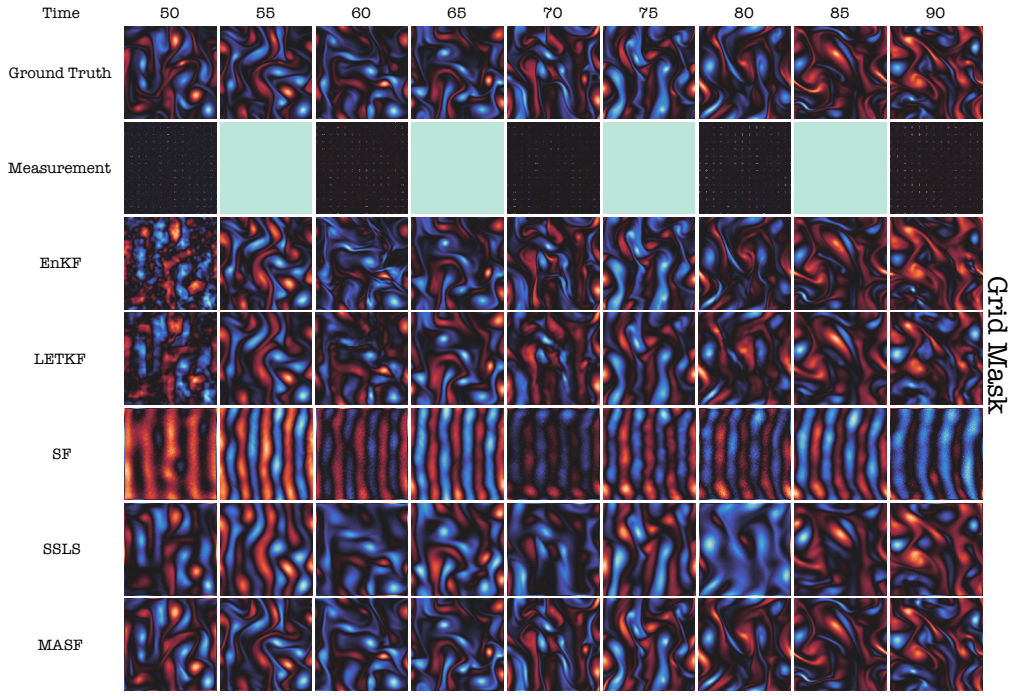


Figure 4: **Grid-mask measurements.** Qualitative comparison of measurements and ensemble means under grid-mask measurements at selected time steps.

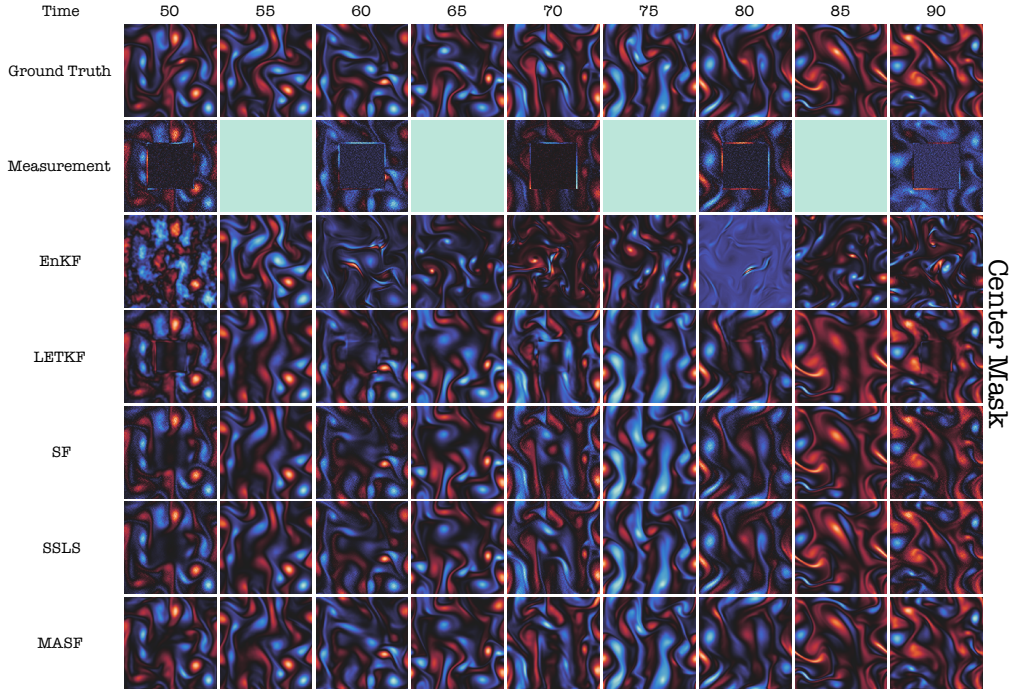


Figure 5: **Center-mask measurements.** Qualitative comparison of measurements and ensemble means under center-mask measurements at selected time steps.

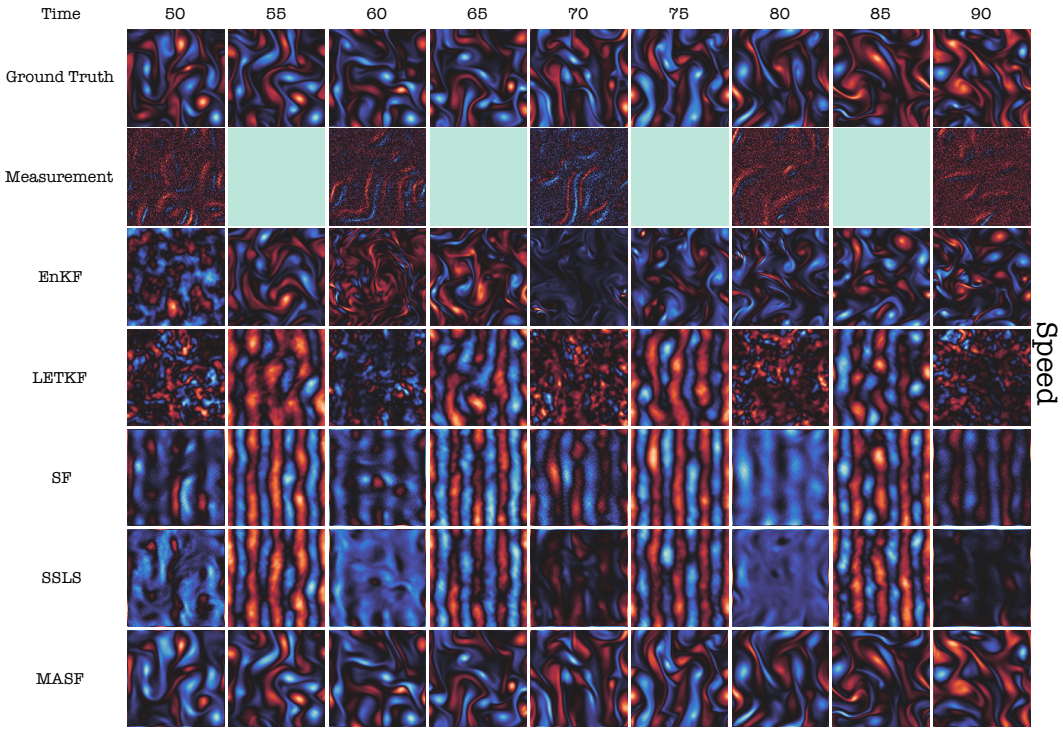


Figure 6: **Speed measurements.** Qualitative comparison of measurements and ensemble means under speed measurements at selected time steps.

D.4 Ensemble-size sensitivity

We analyze how ensemble size affects accuracy and wall-clock time under grid-mask measurements. Figure 7 shows that MASF already reaches a near-saturated accuracy regime with $N = 10$ ensemble members, and increasing the ensemble size yields only marginal RMSE improvement while increasing wall-clock time. In contrast, SF does not improve meaningfully with larger ensembles, and SSLS improves only moderately at substantially higher wall-clock time. For EnKF, the RMSE improves sharply around $N = 250$. LETKF gradually improves as the ensemble size increases, but its wall-clock time remains high even for small ensembles. The RMSE improvement beyond $N = 40$ is marginal, whereas the wall-clock time continues to increase substantially. These results show that MASF achieves a favorable trade-off between accuracy and wall-clock time by maintaining strong accuracy with a small ensemble.

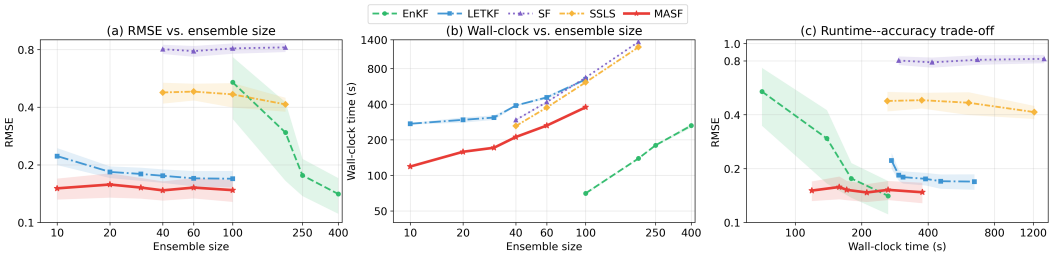


Figure 7: **Ensemble-size sensitivity.** Comparison of RMSE, wall-clock time, and the trade-off between them as functions of ensemble size for the considered methods.

D.5 Extended temporal sensitivity analysis

Figure 8 provides additional results on sensitivity to temporal length and temporal gap under grid-mask measurements. As the assimilation window becomes longer, MASF gradually improves or remains stable, suggesting that repeated measurement updates refine the state estimate. When measurements

become sparser, all methods degrade, but MASF is less sensitive to the temporal gap than the baselines.

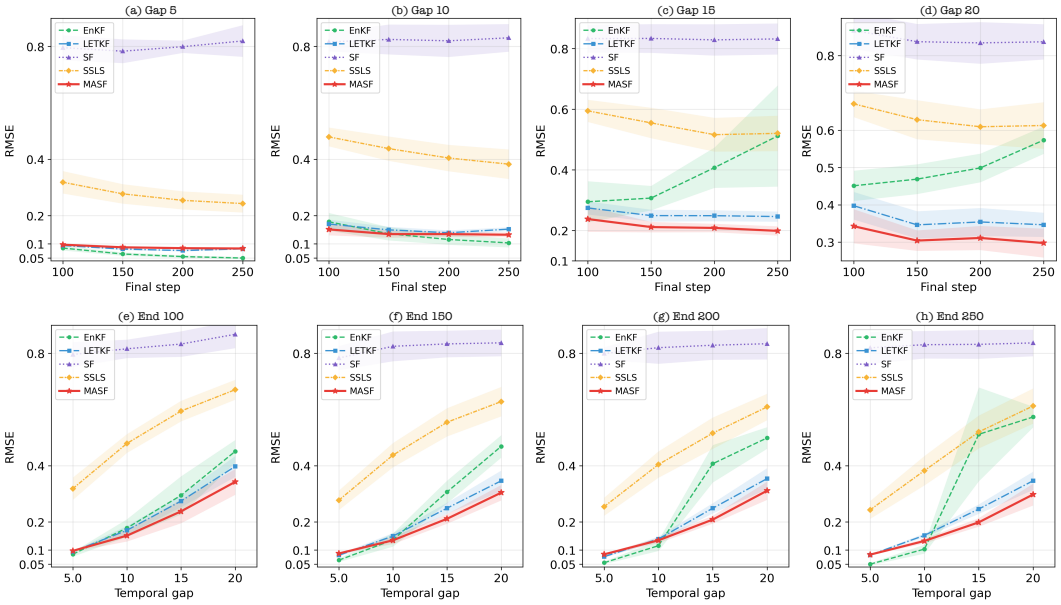


Figure 8: **Extended sensitivity analysis.** Sensitivity to temporal length and temporal gap under grid-mask measurements.

E Ablation studies and cost analysis

This section provides ablation studies and a cost analysis for MASF. We first report component ablations, then examine the effect of pretraining on score-based filtering methods and its total computational cost.

E.1 Component ablations

We conduct component ablations under grid-mask measurements on Kolmogorov flow at 128^2 resolution.

For the main experiment, the selected configuration in Table 10 uses pretrained initialization, $N = 10$ ensemble members, adaptive statistics updates, `attention_resolutions`$= 8$, and $\mathrm{NFE} = 200$. Online fine-tuning uses a learning rate of 5×10^{-5} and 150 full epochs. The assimilation window spans steps 50–100 with a measurement interval of 10 simulation steps. Table 18 reports representative ablations.

Parameter	From	To	RMSE (↓)
Normalization			
`current_config`	–	–	$\mathbf{0.151 \pm 0.019}$
`same_normalization`	`true`	`false`	0.229 ± 0.017
`normalization_form`	`affine`	`scale_only`	0.180 ± 0.013
`stats_mode`	`standard`	`robust`	0.157 ± 0.020
`stats_update_mode`	`moving`	`fixed`	$\mathbf{0.151 \pm 0.018}$
`stats_update_mode`	`moving`	`adaptive`	0.152 ± 0.017

Parameter	From	To	RMSE (↓)
Model selection			
`current_config`	–	–	0.151 ± 0.019
`model_channels`	32	16	0.165 ± 0.020
`num_res_blocks`	2	1	0.167 ± 0.016
`num_res_blocks`	2	3	0.159 ± 0.016
`attention_resolutions`	8	16	$\mathbf{0.147 \pm 0.015}$
Pretraining			
`current_config`	–	–	$\mathbf{0.146 \pm 0.015}$
`batch_size`	16	32	0.169 ± 0.021
`batch_size`	16	64	0.176 ± 0.018
`epoch`	400	100	0.187 ± 0.019
`epoch`	400	200	0.165 ± 0.005
`epoch`	400	300	0.163 ± 0.018
`lr`	0.0003	0.0001	0.184 ± 0.024
`lr`	0.0003	0.0005	0.159 ± 0.021
Sampling			
`current_config`	–	–	0.146 ± 0.015
NFE	200	150	$\mathbf{0.144 \pm 0.018}$
NFE	200	250	0.147 ± 0.018
Fine-tuning			
`current_config`	–	–	$\mathbf{0.144 \pm 0.018}$
`lr`	0.0001	0.0003	0.154 ± 0.015
`online.full_epoch`	120	80	0.151 ± 0.017
`online.full_epoch`	120	90	0.150 ± 0.018
`online.full_epoch`	120	100	0.151 ± 0.018
`online.full_epoch`	120	110	0.148 ± 0.016

Table 18: **Component ablations under grid-mask measurements on Kolmogorov flow at 128^2 resolution.** Mean $\pm$ standard deviation over five seeds. Within each block, `current_config` denotes the configuration before applying the changes listed in the From and To columns.

The ablation results show that normalization, model capacity, and pretraining are the most influential components. Separate normalization or scale-only normalization degrades RMSE, and reducing model capacity or pretraining budget also worsens RMSE. In contrast, sampling and online fine-tuning hyperparameters are relatively stable within the tested ranges.

E.2 Effect of pretraining

We evaluate the effect of pretraining for score-based filtering methods under grid-mask and speed measurements. For each method and measurement setting, we compare scratch training with $N = 100$, scratch training with $N = 10$, and pretrained initialization with $N = 10$. Wall-clock time excludes offline pretraining and measures only the online data assimilation procedure. Table 19 reports the pretraining-ablation results.

Pretraining is most beneficial for MASF in the small-ensemble regime. With $N = 10$, pretraining improves MASF RMSE from 0.40 to 0.15 under grid-mask measurements and from 0.67 to 0.27 under speed measurements. It also reduces the online wall-clock time compared with scratch training at the same ensemble size, from 139.6 s to 118.8 s for grid-mask measurements and from 171.0 s to 123.9 s for speed measurements. SF shows little RMSE improvement from pretraining. For SSLS, RMSE decreases from 0.58 to 0.39 under grid-mask measurements, whereas it changes only from 0.87 to 0.83 under speed measurements.

Measurement	Method	Strategy	N	RMSE (↓)	time (s)
Grid mask	SF	Scratch	100	0.81 ± 0.03	670.6 ± 0.9
		Scratch	10	0.87 ± 0.03	100.9 ± 0.7
		Pretraining	10	0.87 ± 0.05	81.9 ± 0.6
	SSLS	Scratch	100	0.51 ± 0.03	611.5 ± 1.1
		Scratch	10	0.58 ± 0.10	89.2 ± 0.6
		Pretraining	10	0.39 ± 0.04	81.3 ± 12.0
	MASF	Scratch	100	0.25 ± 0.04	535.7 ± 5.2
		Scratch	10	0.40 ± 0.04	139.6 ± 3.6
		Pretraining	10	$\mathbf{0.15 \pm 0.02}$	$\mathbf{118.8 \pm 2.5}$
Speed	SF	Scratch	100	0.89 ± 0.03	380.5 ± 7.1
		Scratch	10	0.95 ± 0.06	112.4 ± 5.0
		Pretraining	10	0.89 ± 0.09	86.8 ± 4.9
	SSLS	Scratch	100	0.79 ± 0.05	358.9 ± 2.2
		Scratch	10	0.87 ± 0.09	97.1 ± 0.9
		Pretraining	10	0.83 ± 0.11	75.5 ± 0.9
	MASF	Scratch	100	0.46 ± 0.06	648.5 ± 1.4
		Scratch	10	0.67 ± 0.08	171.0 ± 12.5
		Pretraining	10	$\mathbf{0.27 \pm 0.04}$	$\mathbf{123.9 \pm 1.0}$

Table 19: **Effect of pretraining under grid-mask and speed measurements on Kolmogorov flow at 128^2 resolution.** Mean $\pm$ standard deviation over five seeds. For each method, scratch training with $N = 100$ and $N = 10$ is compared with pretrained initialization at $N = 10$. Time denotes online wall-clock time.

E.3 Amortized cost analysis

For a fixed state equation and measurement configuration, let T_{pre} be the one-time pretraining cost, T_{MASF} the online cost per assimilation run, and T_B the corresponding per-run cost of baseline B. The break-even number is the smallest positive integer R satisfying

$$T_{\mathrm{pre}} + RT_{\mathrm{MASF}} < RT_B. \tag{137}$$

Table 20 reports the measured offline costs and break-even counts, assuming checkpoint reuse under the same state equation and measurement configuration. The 128^2 grid-mask setting requires 18, 5, 2, and 3 runs relative to EnKF, LETKF, SF, and SSLS, respectively. For the 256^2 and 512^2 grid-mask settings, pretraining is amortized within one or two runs for the available comparisons.

Measurement	Resolution	Pretraining	EnKF	LETKF	SF	SSLS
Grid mask	128^2	18.1 min	18	5	2	3
Grid mask	256^2	18.3 min	2	1	2	2
Grid mask	512^2	20.9 min	1	1	–	–

Table 20: **One-time offline pretraining costs and break-even numbers of assimilation runs, including both offline and online MASF costs.** A count of one means that MASF is already cheaper in total from the first run; dashes indicate unavailable comparisons.

F Reproducibility details

F.1 Compute resources

Table 21 summarizes the compute environment used in our experiments. Wall-clock times for the online data assimilation procedure are reported in the experimental tables. For MASF, the reported

wall-clock time excludes offline pretraining and includes only online fine-tuning and guided sampling. Unless otherwise specified, each model-based run used one GPU, and quantitative results are averaged over five random seeds.

Resource	Specification
Operating system	Linux, kernel 5.14.0
CPU	Intel Xeon 6530P, 2 sockets
CPU capacity	64 physical cores, 128 logical CPUs total
CPU allocation	8 logical CPUs per Slurm task
System memory	250 GiB RAM, 63 GiB swap
GPU	NVIDIA RTX PRO 6000 Blackwell Server Edition
GPU memory	97,887 MiB per GPU
NVIDIA driver	580.105.08
CUDA	13.0, as reported by `nvidia-smi`
PyTorch	2.9.0 with CUDA 13.0 support
JAX / JAX-CFD	JAX 0.4.33, JAX-CFD 0.2.1
Numerical packages	NumPy 2.1.1, SciPy 1.14.1
GPU usage	One GPU per model-based run unless otherwise specified
Default random seeds	5 seeds: 42, 43, 44, 45, 46
Reported wall-clock time	Online data assimilation wall-clock time
MASF wall-clock time	Offline pretraining excluded from reported online wall-clock time

Table 21: **Compute resources and software environment used for the experiments.**

F.2 Existing assets, versions, and licenses

Table 22 summarizes the main existing software and methodological assets used in the experiments. The Kolmogorov flow data used in this paper are generated by the authors through simulation and are not repackaged from an external dataset.

Asset	Use	Version / source	License / terms
JAX-CFD	Kolmogorov flow simulation	0.2.1	Apache-2.0
JAX	Numerical computation	0.4.33	Apache-2.0
PyTorch	Model training	2.9.0	BSD-style license
NumPy	Numerical computation	2.1.1	BSD 3-Clause
SciPy	Numerical computation	1.14.1	BSD 3-Clause
Optuna	Hyperparameter tuning	Cited package	MIT license
U-Net	Architecture reference	Cited paper	–
EnKF / LETKF	Baseline methods	Cited papers	–
SF / SSLS	Score-based baselines	Cited papers	–

Table 22: **Existing assets used in the experiments.**

F.3 Code availability

MASF implementation and configurations are available at `https://github.com/tcnllab-oss/masf`.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the main contributions: a measurement-aware forward process, likelihood-score estimation for linear and nonlinear measurements, and empirical improvements on Lorenz96 and Kolmogorov flow. These claims are supported by the theoretical development in Sections 3.1–3.2, and the experimental results in Sections 4.1–4.4.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitations section discusses the assumptions of known state and measurement equations, sensitivity to misspecification and noisy measurements, the endpoint approximation for nonlinear measurements, online fine-tuning cost, hyperparameter sensitivity, and the limited experimental scope.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The theoretical results state their assumptions, including differentiability, positive semidefiniteness of $\dot{\Sigma}(t)$, Markovian projection regularity, invertibility of $A(t)$ in the linear case, and covariance positive semidefiniteness. Proofs and derivations are provided in Appendix A, with the main text giving the corresponding theorem statements and cross-references.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper specifies the state equations, measurement operators, noise levels, measurement times, baselines, metrics, and random seeds for Lorenz96 and Kolmogorov flow. Appendices B and C provide the algorithm and implementation details, including the Kolmogorov-flow model architectures, tuning ranges, and selected hyperparameters. Additional experiments, ablations and cost analyses, and reproducibility details are provided in Appendices D, E, and F, respectively.

   Guidelines:

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

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The implementation and experimental configurations are publicly available in the project repository, as described in Appendix F.3.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Appendix C.2 specifies the Lorenz96 and Kolmogorov-flow state equations, measurement operators, noise levels, measurement times, ensemble sizes, and Lorenz96 random seeds. The Kolmogorov-flow model architectures, optimization settings, pretraining and online fine-tuning settings, and baseline tuning ranges are given in Appendices C.5–C.8, with the Kolmogorov-flow random seeds specified in Appendix F.1.

   Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: Results are reported as mean $\pm$ standard deviation over 20 seeds (42–61) for Lorenz96 and five seeds (42–46) for Kolmogorov flow, unless otherwise specified. Shaded regions in the sensitivity plots denote standard deviation over five seeds.

Guidelines:

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

8. **Experiments compute resources**

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

Answer: [Yes]

Justification: Online wall-clock times for Kolmogorov flow are reported in the experimental tables, and Appendix E.3 reports offline pretraining costs and an amortized-cost analysis. Appendix F.1 describes the compute resources and software environment. Experiments were run on Linux compute nodes with two Intel Xeon 6530P sockets, 32 cores per socket, 128 logical CPUs, and 250 GiB system memory. GPU experiments used NVIDIA RTX PRO 6000 Blackwell Server Edition GPUs with 97,887 MiB memory per GPU. Unless otherwise specified, each model-based run used one GPU.

Guidelines:

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

9. **Code of ethics**

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

Answer: [Yes]

Justification: The work focuses on algorithmic development and simulation-based experiments for data assimilation. It does not involve human subjects, private data, scraped data, or high-risk model release, and the paper discusses limitations and broader impacts.

Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: The paper includes a broader impacts section. It discusses potential positive impacts for scientific and engineering data assimilation applications, as well as possible negative impacts from incorrect estimates, misspecified dynamics or measurements, biased measurements, and deployment in sensitive monitoring or forecasting systems.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper does not release high-risk models such as language models or image generators, nor does it introduce scraped datasets or assets with high misuse potential. The work is based on simulation-based scientific data assimilation experiments.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The paper credits the main existing scientific and software assets used in the experiments, including JAX-CFD, U-Net, SF and SSLS, and ensemble Kalman filtering methods. Appendix F.2 lists the main software packages, versions, and licenses where applicable. The experiments use simulated Lorenz96 and Kolmogorov flow data generated by the authors rather than a repackaged external dataset.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The public repository provides the implementation and experimental configurations. Algorithms, evaluation metrics, and experimental settings are documented in Appendices B and C, and software versions and licenses in Appendix F.2.

    Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing, user studies, or research with human subjects.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing, user studies, or research with human subjects, so IRB approval is not applicable.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The core methodology does not use LLMs as an important, original, or non-standard component. The proposed method, theory, experiments, and evaluations are based on score-based filtering and simulation-based data assimilation.

Guidelines:

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