GENERATIVE MODEL PROPOSAL BASED PARTICLE FILTERING FOR DATA ASSIMILATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Data assimilation (DA) models state dynamics conditioned on sequential observations, with wide-ranging scientific applications. In filtering, the goal is to model the posterior over the current state given all observations so far. Classical solutions make simplifying distributional or functional assumptions, e.g., linear-Gaussian systems, which can be inaccurate. In principle, particle filters remove these assumptions, yet often collapse in high dimensions. Recent generative DA approaches learn conditional state transitions, but typically incorporate observations through approximate guidance, so errors can accumulate over long horizons. We introduce Flow Proposal Particle Filters (FPPF), which learn a conditional generative-model-based proposal approximating the variance-minimizing optimal proposal. Conditioning on observations steers particles toward high-likelihood regions before weighting, reducing weight variance and delaying degeneracy. Since our proposal admits tractable likelihood evaluation, FPPF computes accurate importance weights and retains a Bayesian update. We further introduce L-FPPF, which uses localization to scale FPPF to high-dimensional systems where global particle filters collapse. Experiments on chaotic dynamical systems and multimodal observations of severe convective storms show that FPPF improves accuracy and calibration over several statistical baselines and other generative DA methods, especially in strongly nonlinear settings, and remains stable over long horizons.

1 INTRODUCTION

Data assimilation (DA) is the inverse problem of inferring the latent state trajectory $\mathbf{x}_{1:T}$ of a dynamical system from imperfect observations $\mathbf{o}_{1:T}$, using knowledge of the system's dynamics $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ and an observation model $p(\mathbf{o}_t \mid \mathbf{x}_t)$. DA is fundamental to many scientific and engineering applications, including weather forecasting (Lorenc, 1986a; Rabier et al., 2000), climate analysis (Dee et al., 2011), agriculture (de Wit & van Diepen, 2007), and motion tracking (Bar-Shalom et al., 2001). In these domains, point estimates of latent states $\mathbf{x}_t$ alone are inadequate. Since downstream decisions based on latent state estimates carry significant consequences, reliable uncertainty quantification requires recovering a full distribution over states.

In this work, we focus on one of the most widely studied DA tasks called *filtering* (Särkkä, 2013; Chen, 2003): at time t, given observations $\mathbf{o}_{1:t}$, we seek the filtering posterior $p(\mathbf{x}_t \mid \mathbf{o}_{1:t})$. Bayesian filtering proceeds via a recursive prediction–update cycle, in which the state dynamics model defines the prior over the next state, and conditioning on the observation using Bayes' rule gives the posterior. When the dynamics and observation models are linear and the noise is Gaussian, this is solved exactly by the Kalman filter (KF) (Kalman, 1960). To handle nonlinear dynamics, models like the Ensemble KF (EnKF) (Evensen, 1994), which are heavily used in operational settings (Houtekamer et al., 2014b), propagate an ensemble through dynamics but retain Gaussian assumptions in the update step.

Particle filters (PFs) (Gordon et al., 1993; Doucet et al., 2000) drop these assumptions by representing the posterior with weighted particles, but suffer from *degeneracy*: particles propagated through the dynamics rarely land where the observation has support, and the ensemble size needed to avoid weight collapse grows exponentially with the effective dimension (Snyder et al., 2008a; Bengtsson et al., 2008). Sequential importance resampling (SIR) (Doucet et al., 2000) allows any proposal $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t)$, corrected by importance weights, and the variance-minimizing *optimal proposal* $q^\star = p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t)$ steers particles toward the observation before weighting (Doucet et al., 2000).

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

However, $q^\star$ is tractable only in special cases, and existing approximations, including look-ahead selection (Pitt & Shephard, 1999), implicit sampling (Chorin & Tu, 2009; Morzfeld et al., 2012), and equal-weight schemes (van Leeuwen, 2010; Ades & van Leeuwen, 2015), rely on restrictive structure or hand-designed surrogates.

Recent generative DA methods instead learn a forecast prior with diffusion or stochastic interpolant models and incorporate observations at inference through likelihood guidance (Rozet & Louppe, 2023; Huang et al., 2024; Chen et al., 2025; Andrae et al., 2026). Guidance requires the likelihood of the observation given a noisy intermediate state, which is intractable and is approximated by Monte Carlo (Chen et al., 2025) or moment matching (Andrae et al., 2026; Savary et al., 2026). These approximations can be inaccurate for strongly nonlinear observation operators and multimodal distributions, and most methods use the guided samples without importance weights, so the error is never corrected. FA-APF (Savary et al., 2026) embeds guidance in an auxiliary particle filter, but its weights evaluate the likelihood at each particle's forecast mean and do not account for mismatch between the guided proposal and the transition density.

In this work, we address this challenge with Flow Proposal Particle Filters (FPPF). The key idea is to learn a proposal that approximates the locally optimal proposal $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t)$, while retaining particle weighting as the "update step". We parameterize such a proposal $q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t)$ with a conditional generative model that is conditioned on observations during training, so no approximate guidance is needed at inference. In practice, we find that flow matching, even with a simple deterministic flow, provides both efficient sampling and tractable likelihood evaluation, enabling us to compute the weight update. Particles are thus proposed in an observation-informed manner and then corrected with importance weights, keeping the generality of particle filtering, with no Gaussian or linearity assumptions, while reducing weight variance and delaying degeneracy. Further, we extend FPPF to high-dimensional systems with a localized variant, L-FPPF, which uses a patch-based velocity network to factorize the full importance weight.

We summarize our main contributions below:

1. FPPF, a particle filter with a learned observation-conditioned flow proposal approximating $q^\star$, whose tractable density gives SIR weights without Gaussian or linear assumptions or guidance.
2. L-FPPF, which localizes the full SIR correction through a factorized proposal log-density, making it suitable and scalable for high-dimensional DA without degeneracy.
3. Experiments on chaotic dynamical systems, including Lorenz-96, Lorenz-63, the surface quasi-geostrophic (SQG) model, and multimodal observations of severe convective storms (SEVIR), showing that FPPF and L-FPPF improve accuracy and calibration over classical and generative baselines, especially in high dimensions and under strongly nonlinear observations.

2 Preliminaries

2.1 Data Assimilation

We consider a dynamical system with state $\mathbf{x}_t \in \mathbb{R}^{d_x}$ and observations $\mathbf{o}_t \in \mathbb{R}^{d_o}$, where t denotes the discrete time index. The evolution of the system is governed by a state transition model $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$, and observations are related to the latent state through the likelihood $p(\mathbf{o}_t \mid \mathbf{x}_t)$. The goal is to estimate the *filtering posterior* $p(\mathbf{x}_t \mid \mathbf{o}_{1:t})$, the distribution over the current state given all observations up to time t. In the Bayesian filtering framework (Jazwinski, 1970; Särkkä, 2013), this distribution is computed recursively via a prediction-update cycle:

$$\begin{aligned} p(\mathbf{x}_t \mid \mathbf{o}_{1:t-1}) &= \int p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} \mid \mathbf{o}_{1:t-1})\, d\mathbf{x}_{t-1}, && \textbf{[PREDICT]} \\ p(\mathbf{x}_t \mid \mathbf{o}_{1:t}) &\propto p(\mathbf{o}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{o}_{1:t-1}). && \textbf{[UPDATE]} \end{aligned} \tag{1}$$

For linear-Gaussian systems, in which the transition and observation models are linear with additive Gaussian noise, the Kalman filter (KF) (Kalman, 1960) solves Eq. (1) in closed form. Without such assumptions, the prediction integral is intractable, and the filtering posterior must be approximated.

2.2 Particle Filters

Particle filters (PFs) approximate the filtering posterior using a weighted ensemble of N particles, $\mathcal{X}_t^w = \{(\mathbf{x}_t^{(i)}, w_t^{(i)})\}_{i=1}^N$, with $\sum_i w_t^{(i)} = 1$. The resulting empirical distribution, $\hat{p}(\mathcal{X}_t^w) =$

$\sum_i w_t^{(i)} \delta_{\mathbf{x}_t^{(i)}}$, converges weakly to $p(\mathbf{x}_t \mid \mathbf{o}_{1:t})$ as $N \to \infty$ under standard conditions. The bootstrap particle filter (BPF) (Gordon et al., 1993) implements the recursion in (1) using Monte Carlo sampling. In the PREDICT step, we start with $\mathcal{X}_{t-1}^w = \{(\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)})\}_{i=1}^N$ representing $p(\mathbf{x}_{t-1}|o_{1:t-1})$, and each particle $\mathbf{x}_{t-1}^{(i)}$ is propagated through the transition dynamics, $\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)})$. In the UPDATE step, each particle receives an unnormalized weight $\tilde{w}_t^{(i)} = w_{t-1}^{(i)}\, p(\mathbf{o}_t \mid \mathbf{x}_t^{(i)})$, which is then normalized as $w_t^{(i)} = \tilde{w}_t^{(i)} / \sum_j \tilde{w}_t^{(j)}$. Thus, the BPF requires only the ability to sample from the transition dynamics $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)})$ and evaluate the observation likelihood $p(\mathbf{o}_t \mid \mathbf{x}_t^{(i)})$.

A central limitation of the BPF is *degeneracy*. After incorporating a new observation, most particles can receive negligible weight, leaving only one or a few particles to represent the posterior. Let $\tau^2 := \mathrm{Var}_i\big[\log p(\mathbf{o}_t \mid \mathbf{x}_t^{(i)})\big]$ denote the log-likelihood variance across the propagated ensemble. Asymptotically, the weights concentrate on a single particle, whose normalized weight tends to 1, unless $\log N \gtrsim \tau^2/2$ (Snyder et al., 2008b; Bengtsson et al., 2008). Since τ^2 typically increases as more informative dimensions are observed, avoiding degeneracy can require an ensemble size exponential in the number of such dimensions. The BPF uses resampling to limit degeneracy, drawing N particles with replacement according to $w_t^{(i)}$ and resetting their weights to $1/N$ (Liu & Chen, 1998). However, resampling can only redistribute existing particles. Once the weights have collapsed onto a few particles, it predominantly replicates them, leading to a loss of diversity known as *sample impoverishment* (Arulampalam et al., 2002).

Proposal distribution. More generally, sequential importance resampling (SIR) (Doucet et al., 2000) draws particles from a proposal $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t)$ and corrects through the importance weights:

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(\mathbf{o}_t \mid \mathbf{x}_t^{(i)}) p(\mathbf{x}_t^{(i)} \mid \mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_t^{(i)} \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{o}_t)}. \tag{2}$$

The BPF is recovered by taking $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t) = p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$, for which the incremental weight reduces to the observation likelihood. A standard result (Doucet et al., 2000) is that the conditional variance of the incremental weight is minimized by the (locally) *optimal proposal*, $q^\star(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t) = p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t)$. Under $q^\star$, $w_t^{(i)} \propto w_{t-1}^{(i)} p(\mathbf{o}_t \mid \mathbf{x}_{t-1}^{(i)})$, so the incremental weight is independent of the sampled $\mathbf{x}_t^{(i)}$. Intuitively, conditioning on the new observation guides particles toward regions consistent with $\mathbf{o}_t$ before weighting, reducing degeneracy. However, outside special cases such as linear-Gaussian models, $q^\star$ is generally intractable, and degeneracy still arises in high dimensions (Snyder et al., 2015).

3 FLOW PROPOSAL PARTICLE FILTERS

We now present Flow Proposal Particle Filter (FPPF), our approach for learning observation-informed proposal distributions that approximate the locally optimal proposal. Recall that BPF propagates particles according to $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$. In contrast, FPPF learns $q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t)$ to approximate $q^\star(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t) = p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t)$. Conditioning the proposal on the incoming observation steers particles toward regions that are more consistent with the filtering update before weighting, reducing weight variance and delaying degeneracy.

3.1 LEARNING THE PROPOSAL

We parameterize the proposal $q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t)$ using conditional flow matching (FM) (Lipman et al., 2023; Liu et al., 2023). FM provides stable training through a simple regression objective, efficient sampling, and importantly, tractable density evaluation for the weight update in Section 3.2.

Let $s \in [0, 1]$ denote flow time, distinct from the discrete filtering index t. A conditional velocity field whose terminal state $\mathbf{z}(1)$ is a sample from $q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t)$ defines the ODE

$$\frac{d\mathbf{z}(s)}{ds} = v_\phi(\mathbf{z}(s), s; \mathbf{x}_{t-1}, \mathbf{o}_t), \qquad \mathbf{z}(0) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{3}$$

Training uses tuples $(\mathbf{x}_{t-1}, \mathbf{o}_t, \mathbf{x}_t)$ from the available trajectory data. Following rectified flow, we draw $\mathbf{z}(0) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, set $\mathbf{z}(1) = \mathbf{x}_t$, sample $s \sim \mathrm{Unif}[0,1]$, and define $\mathbf{z}(s) = (1-s)\mathbf{z}(0) + s\mathbf{z}(1)$. We minimize $\mathcal{L}_{\mathrm{FM}}(\phi) = \mathbb{E}\left[\left\| v_\phi(\mathbf{z}(s), s; \mathbf{x}_{t-1}, \mathbf{o}_t) - \big(\mathbf{z}(1) - \mathbf{z}(0)\big)\right\|_2^2\right]$.

Observation conditioning. Note that we condition the proposal directly on the current observation during training. Recent generative DA methods instead incorporate observations at inference time through likelihood guidance. The popular guidance based approach allows a pretrained generative model to be reused across observation settings, but requires the likelihood score at intermediate flow or diffusion times, which is generally unavailable and must be approximated. Existing approaches estimate such scores using finite-sample Monte Carlo (Chen et al., 2025) or moment matching (Andrae et al., 2026; Savary et al., 2026), introducing sampling error or approximation bias. Moment matching can be particularly inaccurate for multimodal intermediate distributions or highly nonlinear observation operators. Thus, we instead learn the observation-conditioned proposal directly, with no need for inference time guidance. Learning q_ϕ need not require retraining a full model. Given a pretrained generative forecaster, we can freeze its backbone and train only a lightweight adapter that injects $\mathbf{o}_t$ into the velocity field, as in Zhang et al. (2023). During training, we regularize by (1) randomly dropping $\mathbf{o}_t$ as in (Ho & Salimans, 2022), to avoid over-reliance on the observation, and (2) masking coordinates of $\mathbf{x}_{t-1}$ to prevent the proposal network from ignoring $\mathbf{o}_t$ when the dynamics are predictable. Full details are given in Appendix C.2.

3.2 Importance weight computation

Given particles $\{(\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)})\}_{i=1}^N$ approximating $p(\mathbf{x}_{t-1} \mid \mathbf{o}_{1:t-1})$, we sample $\mathbf{x}_t^{(i)} \sim q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{o}_t)$ using Eq. (3) and apply the standard SIR correction (Algorithm 1, line 6):

$$\widetilde{w}_t^{(i)} = w_{t-1}^{(i)} \frac{p(\mathbf{o}_t \mid \mathbf{x}_t^{(i)}) p(\mathbf{x}_t^{(i)} \mid \mathbf{x}_{t-1}^{(i)})}{q_\phi(\mathbf{x}_t^{(i)} \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{o}_t)}. \tag{4}$$

When the transition density $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ is known, FPPF uses it directly. When it is unavailable, or when the dynamics are only available through a simulator from which we can sample but cannot evaluate the transition density, we can learn a probabilistic transition model $\hat{p}_\psi(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ from trajectory data and use its density in the importance correction. FPPF then performs filtering under the learned dynamics model. Similarly, when the observation likelihood $p(\mathbf{o}_t \mid \mathbf{x}_t)$ is unknown, we can learn a probabilistic observation model $\hat{p}_\eta(\mathbf{o}_t \mid \mathbf{x}_t)$ from the data and use its likelihood in the importance correction.

FM provides the proposal density through the instantaneous change-of-variables formula (Lipman et al., 2024): $\log q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t) = \log p_0(\mathbf{z}(0)) - \int_0^1 \mathrm{Tr}(\partial v_\phi / \partial \mathbf{z}(s))\, ds$, where $p_0 = \mathcal{N}(\mathbf{0}, \mathbf{I})$. Computing the exact trace requires d_x backward passes through the network, where d_x is the state dimension, which becomes prohibitive in high dimensions. We instead use the Hutchinson trace estimator, as in prior flow-matching work (Lipman et al., 2024): $\mathrm{Tr}(\mathbf{J}) = \mathbb{E}_\epsilon[\epsilon^\top \mathbf{J} \epsilon]$, where $\mathbb{E}[\epsilon\epsilon^\top] = \mathbf{I}$, and use Rademacher probes, $\epsilon_j \in \{-1, +1\}$. Although the trace estimate is unbiased, exponentiation through the importance ratio introduces additional weight variance; we study this effect in Appendix A.1. Weights are normalized and particles are resampled when the effective sample size $\mathrm{ESS}_t = 1/\sum_i (w_t^{(i)})^2$ falls below a pre-determined threshold N_{eff} (Liu & Chen, 1998). At steps without observations, particles are propagated only through the transition model $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$.

3.3 Comparison to guidance based methods

Here, we briefly compare FPPF with closely related work; an extended discussion and full related work section are given in Appendix A. Closest to our approach, Savary et al. (2026) also approximate $q^\star$ with a generative model, building on the fully adapted auxiliary particle filter (FA-APF) (Pitt & Shephard, 1999), which samples from $q^\star$ and weights by the predictive likelihood $p(\mathbf{o}_t \mid \mathbf{x}_{t-1})$, both intractable in general. Using a pretrained diffusion emulator, they (i) sample the proposal with MMPS guidance, which treats the denoising posterior as Gaussian and linearizes h; (ii) neglect the dependence of its covariance on the noisy state; (iii) evaluate the weight at the forecast mean, $\mathcal{N}(\mathbf{o}_t; h(\mathbb{E}[\mathbf{x}_t \mid \mathbf{x}_{t-1}]), \Sigma_o)$; and (iv) omit any correction for mismatch between the guided sampler

and $q^\star$. DAISI (Andrae et al., 2026) uses similar MMPS guidance, inheriting concerns in (i) and (ii), and forms an unweighted ensemble, so as in (iv) the guidance error is never corrected. FPPF avoids all four of these concerns by learning the conditioning during training and weighting each particle with the full SIR ratio, at the cost of requiring observations at training time.

4 Localized Flow Proposal Particle Filters

An observation-informed proposal reduces weight variance but does not remove the curse of dimensionality; even the optimal proposal can collapse as the effective dimension grows (Snyder et al., 2008b; 2015). For spatial systems, localization offers a way around this. Localized particle filters partition the domain into small blocks and weight each block using only nearby observations (Poterjoy, 2016; Farchi & Bocquet, 2018), so the effective dimension of each update scales with the local neighborhood rather than the full state. These methods use the bootstrap proposal, where the importance weight reduces to the observation likelihood. With independent observation errors, this likelihood factorizes over observations and is localized by keeping only the observations within each block's neighborhood. However, in FPPF, the learned proposal introduces the transition–proposal ratio p/q_ϕ, which must be localized as well. We propose L-FPPF, which does this by constructing spatially local contributions to the proposal density.

Localized flow proposal. We share a velocity network across spatial locations. At location j, $v_{\phi,j}$ depends only on a neighborhood of the evolving flow state $\mathbf{z}(s)$, together with fixed context c_t derived from $\mathbf{x}_{t-1}$ and $\mathbf{o}_t$. This design is motivated by recent work on localized generative models, which found that for distributions with sparse spatial dependencies, local score components can be well approximated from bounded neighborhoods, with localization error decreasing as the neighborhood grows (Gottwald et al., 2025). Although this result is derived for score-based diffusion models, we use the same locality principle as an inductive bias for the flow proposal. The fixed context may contain broader information, such as a global forecast.

For a factorized base distribution, the flow log density can be written as

$$\ell_j^q = \log p_{0,j}(z_j(0)) - \int_0^1 \frac{\partial v_{\phi,j}}{\partial z_j}(\mathbf{z}(s), s; c_t)\, ds, \qquad \log q_\phi = \sum_j \ell_j^q. \tag{5}$$

Because the base distribution factorizes and $\mathrm{Tr}(\partial v_\phi/\partial \mathbf{z}) = \sum_j \partial v_{\phi,j}/\partial z_j$, these terms sum to the joint proposal log density. Their evaluation is also now tractable. Each $v_{\phi,j}$ depends only on a small neighborhood, so $\partial v_{\phi,j}/\partial z_j$ can be computed directly for each location and evaluated in parallel, avoiding the Hutchinson estimator used by global FPPF.

Localized update. We partition the state into disjoint analysis blocks $\{\mathcal{B}_b\}_{b=1}^B$, such as individual grid cells or small spatial patches. Each block receives information from nearby observations through a distance taper, while the transition–proposal correction is restricted to that block. Writing ℓ_m^o and ℓ_j^p for local contributions to the observation and transition log densities, respectively,

$$\log \widetilde{w}_{b,t}^{(i)} = \log w_{b,t-1}^{(i)} + \sum_{m=1}^{d_o} \rho_{b,m}\, \ell_{m,t}^{o,(i)} + \sum_{j\in\mathcal{B}_b} \left(\ell_{j,t}^{p,(i)} - \ell_{j,t}^{q,(i)}\right), \tag{6}$$

where $\rho_{b,m} = G(d(b,m)/r_{\mathrm{loc}})$ is a spatial taper. Observation neighborhoods may overlap even though the analysis blocks are disjoint.

Weights are normalized and resampled independently within each block, and full-state particles are reconstructed from the selected block ancestors (Farchi & Bocquet, 2018). This replaces one high-dimensional importance update with many low-dimensional ones while retaining the learned observation-informed proposal. The construction is a localized approximation to the global SIR update; it does not assume that the filtering posterior factorizes across blocks.

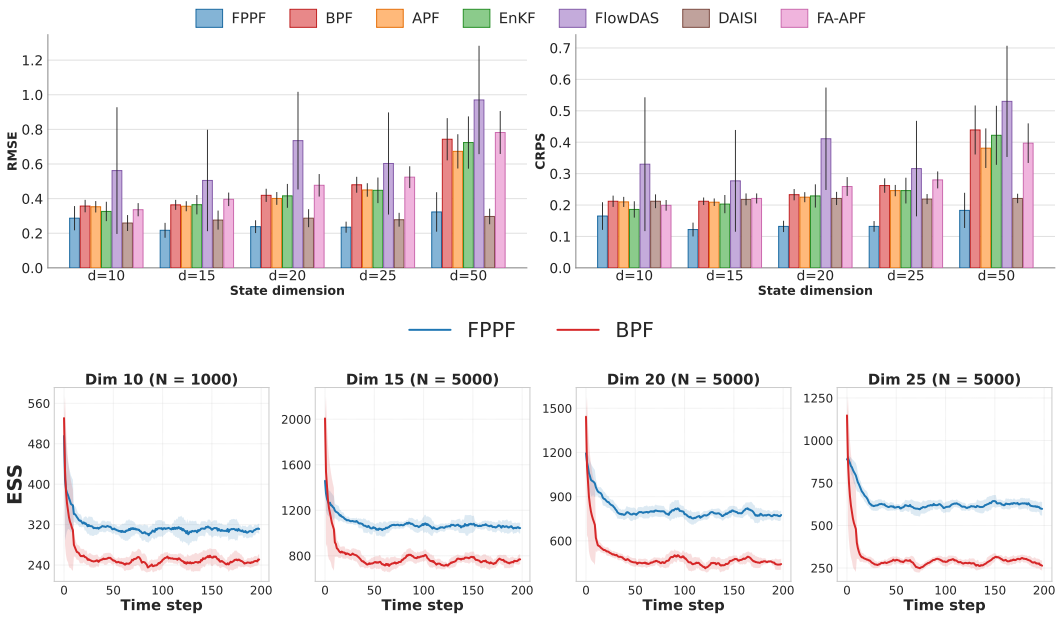


Figure 1: Lorenz-96 scaling with $h(\mathbf{x}) = \arctan(\mathbf{x})$. **Top:** RMSE and CRPS. Bars show means over trajectories and error bars one standard deviation. FPPF (blue) achieves the lowest CRPS at every d_x and the lowest RMSE at $d_x = 15, 20, 25$; DAISI achieves the lowest RMSE at $d_x = 10, 50$. Full results are in the appendix. **Bottom:** pre-resampling ESS over time, averaged over trajectories. FPPF maintains higher ESS than BPF, but global importance sampling still degrades with dimension.

5 Experiments

We evaluate FPPF in four regimes: (1) scaling with state dimension (2) localized filtering in high-dimensional spatial systems, (3) strongly nonlinear, non-Gaussian filtering, and (4) real-world weather data assimilation.

5.1 Scaling with state dimension

Lorenz-96 (Lorenz, 1996) is a chaotic model of atmospheric dynamics on a periodic domain whose dimensionality can be changed without affecting local dynamics, making it ideal to study how filtering degrades with dimension. We take state dimension $d_x \in \{5, 10, 15, 20, 25, 50\}$ and observe the full state through $h(\mathbf{x}) = \arctan(\mathbf{x})$, with Gaussian process and observation noise of standard deviation 0.2. All PFs use $N = 1000$ particles at $d_x = 10$ and $N = 5000$ at $d_x = 15, 20, 25$. For each d_x we train a separate flow proposal on simulated trajectories, and report RMSE and CRPS averaged over 100 trajectories of 200 steps. We compare against the bootstrap PF (BPF; Gordon et al., 1993), auxiliary PF (APF; Pitt & Shephard, 1999), and ensemble Kalman filter (EnKF; Evensen, 1994), and against three generative DA methods that condition on observations at inference time: FlowDAS (Chen et al., 2025), DAISI (Andrae et al., 2026), and FA-APF (Savary et al., 2026).

FPPF achieves the lowest CRPS at every dimension and the lowest RMSE at $d_x = 15, 20, 25$ (Figure 1). Its advantage over BPF grows with d_x and is largest at $d_x = 50$. This isolates the effect of learning the proposal. BPF samples from the transition prior, so as d_x grows fewer particles are consistent with the observation and the weights degenerate, whereas FPPF conditions its proposal on the observation. DAISI is competitive in RMSE, achieving the lowest values at $d_x = 10, 50$, but has higher CRPS than FPPF throughout. FA-APF improves on BPF at small d_x and in CRPS in some settings, but its RMSE falls behind BPF as d_x grows. APF improves on BPF at larger d_x without closing the gap to FPPF, and EnKF is comparable to the classical baselines. Proposal learning delays, but does not solve, the curse of dimensionality (Section 4). FPPF maintains higher pre-resampling effective sample size (ESS) than BPF, with the gap widening in d_x (Figure 1, bottom), but its absolute ESS still becomes small at high dimension. Recovering it by increasing the particle count is prohibitively expensive, which motivates the localized construction studied next.

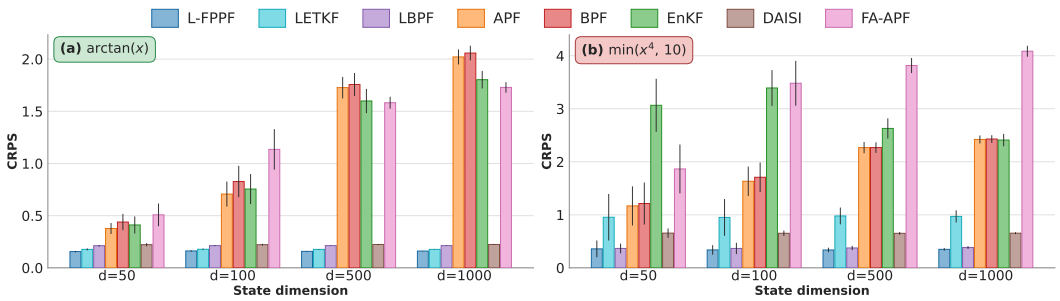


Figure 2: CRPS on high-dimensional Lorenz-96 for localized and global filters under (a) $h(\mathbf{x}) = \arctan(\mathbf{x})$ and (b) $h(\mathbf{x}) = \min(\mathbf{x}^4, 10)$. L-FPPF (blue) achieves the lowest CRPS at every d_x. Bars show means over trajectories and error bars one standard deviation. Full results are in Appendix F.

5.2 Localized filtering in high-dimensional spatial systems

We determine whether localization allows FPPF to scale to high-dimensional spatial systems, again on Lorenz-96, where the dimension can grow while the local dynamics stay fixed, and then on the surface quasi-geostrophic (SQG) model (Tulloch & Smith, 2009), a more realistic turbulent system.

Lorenz-96. We evaluate L-FPPF at $d_x \in \{50, 100, 500, 1000\}$, where global particle filters become impractical. The local proposal is trained once on the $d_x = 25$ system and applied zero-shot at every target dimension. We compare against two established localized filters: LETKF (Hunt et al., 2007), which performs the EnKF analysis within local neighborhoods, and LBPF (Poterjoy, 2016; Farchi & Bocquet, 2018), which localizes the particle filter weighting and resampling. All three use localization radius $r = 4$; the localized particle filters use $N = 500$ particles. We also report the global BPF, APF, and EnKF, together with the SOTA ML baselines DAISI and FA-APF.

Figure 2(a) shows a clear separation between global and localized filtering. The errors of BPF, APF, EnKF, and FA-APF grow substantially with d_x, whereas LBPF, LETKF, and L-FPPF remain nearly constant, since localization replaces one high-dimensional update with local updates whose effective dimension is set by the localization radius. DAISI, which uses no importance weights, also stays flat. Among the localized methods, LETKF performs well under the $\arctan$ operator, but L-FPPF achieves lower RMSE than both LETKF and LBPF at every dimension, and the lowest CRPS of all methods throughout. It also has the lowest RMSE at three of the four dimensions, with DAISI best at $d_x = 100$ (Appendix F). Localization thus provides the main scaling gain, and the learned local proposal further improves the particle approximation within each local update.

Surface quasi-geostrophic (SQG) model. We next test localization on the SQG model, a two-dimensional turbulent system that evolves potential temperature under nonlinear advection, large-scale forcing, and small-scale dissipation. Its chaotic, multiscale dynamics make it substantially harder than Lorenz-96, and it is a standard benchmark for high-dimensional DA (Liang et al., 2026; Andrae et al., 2026). Following Andrae et al. (2026), we use a 64×64 grid ($d_x = 4096$) and assimilate observations every 3 hours for 100 steps, reporting performance over the last 50. We observe 25% of grid points with $\sigma_{\text{obs}} = 5$, so observations are both spatially sparse and weakly informative. Observation locations are sampled uniformly at random and held fixed within each trajectory. LETKF performs best (Figure 3, left). This is expected, since the observation operator is a linear selection of grid points with Gaussian noise, matching the assumptions of the Kalman update. L-FPPF is second, closely followed by LBPF, and all three localized filters outperform the global and generative baselines by a wide margin. DAISI's analysis is also visibly noisier than L-FPPF's (Figure 3, right). Together with Section 5.3, where LETKF degrades under a nonlinear operator, this shows that L-FPPF is generally applicable method which remains competitive when Gaussian methods are well specified and is more robust when they are not.

5.3 Nonlinear and non-Gaussian filtering regimes

Nonlinear observations and chaotic dynamics can produce filtering distributions that Gaussian approximations describe poorly. These regimes challenge both ensemble Kalman updates and the

Method	RMSE ↓	CRPS ↓
BPF	10.05 ± 0.61	7.99 ± 0.52
LBPF	3.27 ± 0.19	1.81 ± 0.10
LETKF	**2.86** ± 0.22	**1.53** ± 0.11
DAISI	4.51 ± 0.12	2.70 ± 0.08
FA-APF	8.55 ± 0.75	6.24 ± 0.61
L-FPPF	<u>3.24</u> ± 0.20	<u>1.77</u> ± 0.11

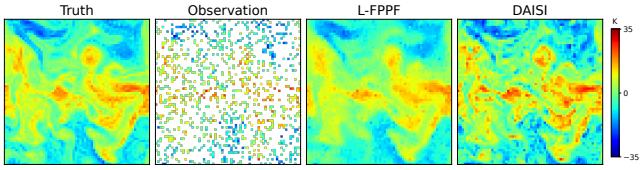


Figure 3: SQG (64×64) with 25% of grid points observed, $\sigma_{\text{obs}} = 5$. **Left:** mean RMSE and CRPS; best in **bold**, second <u>underlined</u>. **Right:** potential temperature at the final assimilation step.

approximations used to condition generative models on observations. FA-APF and DAISI guide sampling with MMPS, which requires the likelihood of the observation given a noisy intermediate state. Since this is intractable, MMPS treats the clean state given the noisy state as Gaussian and linearizes the observation operator around its mean. The learned prior still allows non-Gaussian samples, but the guidance can be inaccurate when the clean state given the noisy state is multimodal, as is typical at high noise levels (Savary et al., 2026), or when the observation operator is strongly nonlinear. FA-APF adds a second approximation in its particle weights, which evaluate the likelihood at each particle's forecast mean. We test both failure modes below.

Nonlinear observations. We repeat the high-dimensional Lorenz-96 experiment of Section 5.2 with $h(\mathbf{x}) = \min(\mathbf{x}^4, 10)$, applied elementwise. This operator cannot distinguish states of equal magnitude and opposite sign, and saturates for $|x_j| > 10^{1/4}$. L-FPPF achieves the lowest RMSE and CRPS at every dimension, with LBPF second (Figure 2(b); Appendix F). Both weight particles by the full likelihood, making no linear or Gaussian assumption about the observation operator (Poterjoy, 2016). LETKF instead performs a linear update of the state, which is biased for strongly nonlinear operators (Kurosawa & Poterjoy, 2021). FA-APF incurs large errors that grow with dimension. Since DAISI uses similar guidance and degrades far less, we attribute the gap mainly to the FA-APF weights. These evaluate the likelihood at each particle's forecast mean, which for a strongly nonlinear operator can differ substantially from the mean likelihood, and temper it with an exponent chosen to keep the ESS in a fixed range. Because the likelihood is a product over observed components, this exponent must typically shrink as dimension grows, weakening the influence of the observations on particle selection and biasing the filtering distribution. DAISI has no weights or resampling, and starts guided sampling from partially noised forecast members rather than pure noise, avoiding the high noise levels where MMPS is least accurate. Its error is accordingly nearly constant in d_x, but much higher than under $h(\mathbf{x}) = \arctan(\mathbf{x})$. L-FPPF instead learns the proposal directly, without linearizing the observation operator, and corrects for proposal error with localized importance weights.

Multimodal forecast distributions. We next consider Lorenz-63 (Lorenz, 1963), a 3D system whose trajectories switch irregularly between two attractor lobes. We vary the observation interval $K \in \{1, 50, 100\}$ with $h(\mathbf{x}) = \arctan(\mathbf{x})$ and $\sigma_{\text{obs}} = 0.25$. Longer intervals let the forecast spread across both lobes before the next observation, making it bimodal (Figure 4, right). We use $N = 500$ particles and evaluate over 50 trajectories. Methods that rely on Gaussian approximations degrade as K grows (Figure 4, left). EnKF, which assumes a Gaussian forecast, is competitive at $K = 1$ but falls far behind at $K = 50$ and 100. The MMPS-based FA-APF is second at $K = 1$ but falls behind the particle filters at longer intervals, and DAISI trails throughout. BPF, APF, and FPPF, whose updates make no Gaussian assumption, are the three best methods at $K = 50$ and 100, with neglible performance differences (in this very low-dimensional setting there is not much to be gained from the learned proposal). FPPF thus retains the robustness of classical PFs to multimodal forecasts.

5.4 Cross-modal weather assimilation

Here we use real-world data, where neither the dynamics nor the observation model is known. SEVIR (Veillette et al., 2020) contains over 10,000 severe weather events across the continental United States, such as thunderstorms, hail, and tornadoes, with satellite, radar, and lightning imagery. We take the 128×128 vertically integrated liquid (VIL) radar field at 10-minute intervals as the state. Previous work (Andrae et al., 2026) observes VIL directly, assuming sparse but exact measurements of the state. However, in many practical DA settings, the quantity of interest is not observed directly; satellites,

Method	$K = 1$	$K = 50$	$K = 100$
BPF	0.750 $\pm$ 0.122	2.751 $\pm$ 0.443	4.435 $\pm$ 0.872
APF	0.754 $\pm$ 0.124	2.747 $\pm$ 0.451	4.424 $\pm$ 0.853
EnKF	0.800 $\pm$ 0.138	4.266 $\pm$ 0.645	5.818 $\pm$ 0.924
DAISI	2.845 $\pm$ 0.271	3.653 $\pm$ 0.616	5.147 $\pm$ 0.655
FA-APF	0.703 $\pm$ 0.120	2.936 $\pm$ 0.463	4.634 $\pm$ 0.714
FPPF	**0.622** $\pm$ **0.113**	**2.746** $\pm$ **0.441**	**4.409** $\pm$ **0.880**

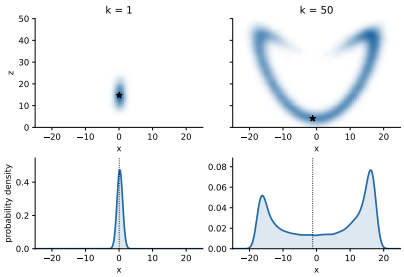


Figure 4: Lorenz-63 with observations every K steps, $\sigma_{\text{obs}} = 0.25$. **Left:** RMSE over 1000-step trajectories. **Right:** an unobserved forecast after $K = 1$ and $K = 50$ steps; by $K = 50$ it is bimodal. Top: (x, z) marginal density; bottom: x marginal.

Method	RMSE $\downarrow$	CRPS $\downarrow$
LETKF	0.193 ± 0.035	0.131 ± 0.027
LBPF	0.159 ± 0.027	0.102 ± 0.022
L-FPPF	$\mathbf{0.151 \pm 0.026}$	$\mathbf{0.097 \pm 0.021}$

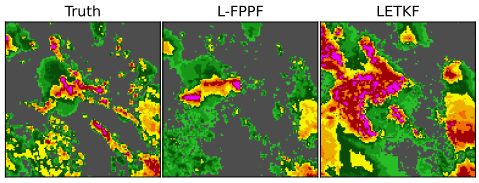


Figure 5: SEVIR cross-modal assimilation. **Left:** mean RMSE and CRPS; best in **bold**, second underlined. **Right:** VIL at the final assimilation step.

for example, provide dense measurements of radiances rather than precipitation. We construct such a scenario by observing two infrared channels, which measure cloud-top and upper-tropospheric water vapor brightness temperatures. Both are resampled to the VIL grid, so each observation $\mathbf{o}_t \in \mathbb{R}^{2\times128\times128}$ covers the full domain, yet its relation to VIL is unknown and highly nonlinear.

We learn all components from data. A rectified flow transition model $\hat{p}_\psi(\mathbf{x}_t \mid \mathbf{x}_{t-1})$, a rectified flow likelihood $\hat{p}_\eta(\mathbf{o}_t \mid \mathbf{x}_t)$, and an infrared-conditioned proposal $q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t)$ trained on top of the frozen transition model. L-FPPF weights particles by $\hat{p}_\eta \hat{p}_\psi / q_\phi$, LBPF samples from $\hat{p}_\psi$ and weights by $\hat{p}_\eta$. LETKF uses the same transition and assumes Gaussian distributed observations. We run 20 assimilation steps on 10 test storms and average metrics over the last 10. We do not compare with DAISI or FA-APF, as their implementations do not support a learned observation model without substantial adaptation. L-FPPF achieves the lowest RMSE and CRPS (Table 5), improving substantially on LETKF. It also improves on LBPF, and since the two share the same learned transition and likelihood, the gain comes from the proposal. The experiment also illustrates the modularity of the framework: learned dynamics, observation, and proposal models combine, and a lightweight observation adapter can be added to an existing forecast model to form a proposal without retraining it.

6 Conclusion and Future Work

We introduced Flow Proposal Particle Filters (FPPF), which learn a conditional generative model approximation to the optimal proposal $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t)$ and correct it with importance weights computed from its tractable density. Classical methods such as the EnKF rely on Gaussian and linear approximations, and guided generative methods rely on approximate likelihood guidance that can degrade under strongly nonlinear observations and multimodal forecasts. FPPF instead corrects proposal error with importance weights, and a deterministic rectified flow suffices, without the stochastic dynamics of diffusion or interpolant-based methods. The framework is also modular; learned forecast or observation models can replace unknown densities, and lightweight observation adapters can be learned from historical data. We further propose L-FPPF, the first localized ML-based DA method, which uses generative model properties to localize the full SIR correction. Across chaotic systems, both methods improve accuracy and calibration over classical and generative baselines, most clearly in high dimensions and under strongly nonlinear observations. Open directions include scaling to operational settings, and reducing inference cost through few-step distillation (Song et al., 2023; Geng et al., 2025; Frans et al., 2025) and fast divergence estimation (Ai et al., 2026).

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for [tasks with required disclosure]. We have not used generative AI tools for [other tasks with required disclosure], and [the rest of the required disclosure tasks] are not applicable to this work. Additionally, we used generative AI tools for [tasks with recommended disclosure]. We have reviewed all AI-assisted work. [Elaborate. For example, "we checked LLM-generated research ideas for potential plagiarism through a manual literature survey", "LLM-generated code was verified and tested for correctness by 2 authors", etc.]. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

See the ICLR 2027 AI Policy for Authors for more details. This statement should not be more than 1 page.

AI USE STATEMENT

We used generative AI tools to assist with code and experiment implementation and debugging, and manuscript preparation. We reviewed the AI-assisted text and analyses and tested the resulting code. We take responsibility for all content and results presented in this work.

REPRODUCIBILITY STATEMENT

We provide the code for our experiments at `https://anonymous.4open.science/r/fppf_supp_code-8012/`. The method, experiments, appendix sections describe the filtering algorithms, dynamical systems, observation models, and evaluation settings.

REFERENCES

Melanie Ades and Peter Jan van Leeuwen. The equivalent-weights particle filter in a high-dimensional system. *Quarterly Journal of the Royal Meteorological Society*, 141(687):484–503, 2015. doi: 10.1002/qj.2370.

Xinyue Ai, Yutong He, Albert Gu, Ruslan Salakhutdinov, J. Zico Kolter, Nicholas Matthew Boffi, and Max Simchowitz. Joint distillation for fast likelihood evaluation and sampling in flow-based models. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL `https://openreview.net/forum?id=8uZ5UdIul2`. arXiv:2512.02636.

Martin Andrae, Erik Larsson, So Takao, Tomas Landelius, and Fredrik Lindsten. Daisi: Data assimilation with inverse sampling using stochastic interpolants. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. URL `https://arxiv.org/abs/2512.00252`.

M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim 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.

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, December 2024. ISSN 0045-7825. doi: 10.1016/j.cma.2024.117447.

Yaakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. John Wiley & Sons, New York, 2001. ISBN 9780471416555. doi: 10.1002/0471221279.

Thomas Bengtsson, Peter Bickel, and Bo Li. *Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems*, pp. 316–334. Institute of Mathematical Statistics, 2008. ISBN 0940600749. doi: 10.1214/193940307000000518. URL `http://dx.doi.org/10.1214/193940307000000518`.

Craig H. Bishop, Brian J. Etherton, and Sharanya J. Majumdar. Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects. *Monthly Weather Review*, 129(3):420–436, 2001. doi: 10.1175/1520-0493(2001)129<0420:ASWTET>2.0.CO;2.

Marc Bocquet, Julien Brajard, Alberto Carrassi, and Laurent Bertino. Bayesian inference of chaotic dynamics by merging data assimilation, machine learning and expectation-maximization. *Foundations of Data Science*, 2(1):55–80, 2020. doi: 10.3934/fods.2020004.

Julien Brajard, Alberto Carrassi, Marc Bocquet, and Laurent Bertino. Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: A case study with the Lorenz 96 model. *Journal of Computational Science*, 44:101171, 2020. doi: 10.1016/j.jocs.2020.101171.

Nicola Branchini and Víctor Elvira. Optimized auxiliary particle filters: adapting mixture proposals via convex optimization. In *Conference on Uncertainty in Artificial Intelligence*, 2020. URL https://api.semanticscholar.org/CorpusID:235446306.

Siyi Chen, Yixuan Jia, Qing Qu, He Sun, and Jeffrey A Fessler. FlowDAS: A stochastic interpolant-based framework for data assimilation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=1nWqhiulqD.

Xiongjie Chen and Yunpeng Li. An overview of differentiable particle filters for data-adaptive sequential Bayesian inference. *Foundations of Data Science*, 2023. doi: 10.3934/fods.2023014.

Xiongjie Chen and Yunpeng Li. Normalizing flow-based differentiable particle filters. *IEEE Transactions on Signal Processing*, 2024. arXiv:2403.01499.

Yuming Chen, Daniel Sanz-Alonso, and Rebecca Willett. Autodifferentiable ensemble Kalman filters. *SIAM Journal on Mathematics of Data Science*, 4(2):801–833, 2022. doi: 10.1137/21M1434477.

Zhe Chen. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182(1): 1–69, 2003. doi: 10.1080/02331880309257.

Alexandre J. Chorin and Xuemin Tu. Implicit sampling for particle filters. *Proceedings of the National Academy of Sciences*, 106(41):17249–17254, 2009. doi: 10.1073/pnas.0909196106.

Adrien Corenflos, James Thornton, George Deligiannidis, and Arnaud Doucet. Differentiable particle filtering via entropy-regularized optimal transport. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2100–2111. PMLR, 2021.

Julien Cornebise, Éric Moulines, and Jimmy Olsson. Adaptive methods for sequential importance sampling with application to state space models. *Statistics and Computing*, 18(4):461–480, 2008. doi: 10.1007/s11222-008-9089-4.

P. Courtier, J.-N. Thépaut, and A. Hollingsworth. A strategy for operational implementation of 4D-Var, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society*, 120(519):1367–1387, 1994. doi: 10.1002/qj.49712051912.

A. J. W. de Wit and C. A. van Diepen. Crop model data assimilation with the Ensemble Kalman Filter for improving regional crop yield forecasts. *Agricultural and Forest Meteorology*, 146(1–2): 38–56, 2007. doi: 10.1016/j.agrformet.2007.05.004.

D. P. Dee, S. M. Uppala, A. J. Simmons, P. Berrisford, P. Poli, S. Kobayashi, U. Andrae, M. A. Balmaseda, G. Balsamo, P. Bauer, P. Bechtold, A. C. M. Beljaars, L. van de Berg, J. Bidlot, N. Bormann, C. Delsol, R. Dragani, M. Fuentes, A. J. Geer, L. Haimberger, S. B. Healy, H. Hersbach, E. V. Hólm, L. Isaksen, P. Kållberg, M. Köhler, M. Matricardi, A. P. McNally, B. M. Monge-Sanz, J.-J. Morcrette, B.-K. Park, C. Peubey, P. de Rosnay, C. Tavolato, J.-N. Thépaut, and F. Vitart. The ERA-Interim reanalysis: configuration and performance of the data assimilation system. *Quarterly Journal of the Royal Meteorological Society*, 137(656):553–597, 2011. doi: 10.1002/qj.828.

Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods. *Journal of Geophysical Research: Oceans*, 99(C5):10143–10162, 1994. doi: 10.1029/94JC00572. URL https://doi.org/10.1029/94JC00572.

Alban Farchi and Marc Bocquet. Review article: Comparison of local particle filters and new implementations. *Nonlinear Processes in Geophysics*, 25(4):765–807, 2018. doi: 10.5194/npg-25-765-2018.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.12557.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, volume 38, 2025. arXiv:2505.13447.

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.

Georg A. Gottwald, Shuigen Liu, Youssef Marzouk, Sebastian Reich, and Xin T. Tong. Localized diffusion models. *arXiv preprint arXiv:2505.04417*, 2025. doi: 10.48550/arXiv.2505.04417.

Shixiang Shane Gu, Zoubin Ghahramani, and Richard E. Turner. Neural adaptive sequential Monte Carlo. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

P. L. Houtekamer, X. Deng, H. L. Mitchell, S.-J. Baek, and N. Gagnon. Higher resolution in an operational ensemble Kalman filter. *Monthly Weather Review*, 142:1143–1162, 2014a.

P. L. Houtekamer, Xingxiu Deng, Herschel L. Mitchell, Seung-Jong Baek, and Normand Gagnon. Higher resolution in an operational ensemble kalman filter. *Monthly Weather Review*, 142(3):1143 – 1162, 2014b. doi: 10.1175/MWR-D-13-00138.1. URL https://journals.ametsoc.org/view/journals/mwre/142/3/mwr-d-13-00138.1.xml.

Langwen Huang, Lukas Gianinazzi, Yuejiang Yu, Peter Dominik Dueben, and Torsten Hoefler. DiffDA: a diffusion model for weather-scale data assimilation. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=vhMq3eAB34.

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(1–2):112–126, 2007. doi: 10.1016/j.physd.2006.11.008.

Andrew H. Jazwinski. *Stochastic Processes and Filtering Theory*, volume 64 of *Mathematics in Science and Engineering*. Academic Press, New York, 1970.

Rico Jonschkowski, Divyam Rastogi, and Oliver Brock. Differentiable particle filters: End-to-end learning with algorithmic priors. In *Proceedings of Robotics: Science and Systems (RSS)*, Pittsburgh, Pennsylvania, 2018. doi: 10.15607/RSS.2018.XIV.001.

Rudolf Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. doi: 10.1115/1.3662552.

Peter Karkus, David Hsu, and Wee Sun Lee. Particle filter networks with application to visual localization. In *Proceedings of the 2nd Conference on Robot Learning (CoRL)*, volume 87 of *Proceedings of Machine Learning Research*, pp. 169–178. PMLR, 2018.

Kenta Kurosawa and Jonathan Poterjoy. Data assimilation challenges posed by nonlinear operators: A comparative study of ensemble and variational filters and smoothers. *Monthly Weather Review*, 149(7):2369–2389, 2021. doi: 10.1175/MWR-D-20-0368.1.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 382(6677): 1416–1421, 2023. doi: 10.1126/science.adi2336. URL https://www.science.org/doi/abs/10.1126/science.adi2336.

Steffen L. Lauritzen. *Graphical Models*. Number 17 in Oxford Statistical Science Series. Clarendon Press, Oxford, 1996.

Tuan Anh Le, Maximilian Igl, Tom Rainforth, Tom Jin, and Frank Wood. Auto-encoding sequential Monte Carlo. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJ8c3f-0b.

Siming Liang, Hoang Tran, Feng Bao, Hristo G. Chipilski, Peter Jan van Leeuwen, and Guannan Zhang. Ensemble score filter with image inpainting for data assimilation in tracking Surface Quasi-Geostrophic dynamics with partial observations. *SIAM Journal on Applied Mathematics*, 86 (5):2379–2404, 2026. doi: 10.1137/25M1730053.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code, 2024. URL https://arxiv.org/abs/2412.06264.

Jun S. Liu and Rong Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998. doi: 10.1080/01621459.1998.10473765.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

A. C. Lorenc. Analysis methods for numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 112(474):1177–1194, 1986a. doi: 10.1002/qj.49711247414.

A. C. Lorenc. Analysis methods for numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 112(474):1177–1194, 1986b.

Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2): 130–141, 1963. doi: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2.

Edward N. Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on Predictability*, volume 1, pp. 1–18, Reading, UK, 1996. ECMWF.

Chris J. Maddison, Dieterich Lawson, George Tucker, Nicolas Heess, Mohammad Norouzi, Andriy Mnih, Arnaud Doucet, and Yee Whye Teh. Filtering variational objectives. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Matthias Morzfeld, Xuemin Tu, Ethan Atkins, and Alexandre J. Chorin. A random map implementation of implicit filters. *Journal of Computational Physics*, 231(4):2049–2066, 2012. doi: 10.1016/j.jcp.2011.11.022.

Christian A. Naesseth, Scott W. Linderman, Rajesh Ranganath, and David M. Blei. Variational sequential Monte Carlo. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 84 of *Proceedings of Machine Learning Research*, pp. 968–977. PMLR, 2018.

Brooks Paige and Frank Wood. Inference networks for sequential Monte Carlo in graphical models. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 3040–3049. PMLR, 2016.

Stephen G. Penny and Takemasa Miyoshi. A local particle filter for high-dimensional geophysical systems. *Nonlinear Processes in Geophysics*, 23(6):391–405, 2016. doi: 10.5194/npg-23-391-2016.

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.

Jonathan Poterjoy. A localized particle filter for high-dimensional nonlinear systems. *Monthly Weather Review*, 144(1):59–76, 2016. doi: 10.1175/MWR-D-15-0163.1.

F. Rabier, H. Järvinen, E. Klinker, J.-F. Mahfouf, and A. Simmons. The ECMWF operational implementation of four-dimensional variational assimilation. I: Experimental results with simplified physics. *Quarterly Journal of the Royal Meteorological Society*, 126(564):1143–1170, 2000. doi: 10.1256/smsqj.56414.

François Rozet and Gilles Louppe. Score-based data assimilation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=VUvLSnMZdX.

Simo Särkkä. *Bayesian Filtering and Smoothing*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, Cambridge, 2013. doi: 10.1017/CBO9781139344203.

Thomas Savary, François Rozet, and Gilles Louppe. Training-free bayesian filtering with generative emulators. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2605.20028.

C. Schraff, H. Reich, A. Rhodin, A. Schomburg, K. Stephan, A. Periáñez, and R. Potthast. Kilometre-scale ensemble data assimilation for the COSMO model (KENDA). *Quarterly Journal of the Royal Meteorological Society*, 142:1453–1472, 2016.

Phillip Si and Peng Chen. Latent-enSF: A latent ensemble score filter for high-dimensional data assimilation with sparse observation data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=urcEYsZOBz.

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

Chris Snyder, Thomas Bengtsson, Peter Bickel, and Jeffrey Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640, 2008b. doi: 10.1175/2008MWR2529.1.

Chris Snyder, Thomas Bengtsson, and Matthias Morzfeld. Performance bounds for particle filters using the optimal proposal. *Monthly Weather Review*, 143(11):4750–4761, 2015. doi: 10.1175/MWR-D-15-0144.1.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 2023.

Taos Transue, Bohan Chen, So Takao, and Bao Wang. Flow matching for efficient and scalable data assimilation. *arXiv preprint arXiv:2508.13313*, 2025. doi: 10.48550/arXiv.2508.13313. URL https://arxiv.org/abs/2508.13313.

Ross Tulloch and K. Shafer Smith. Quasigeostrophic turbulence with explicit surface dynamics: Application to the atmospheric energy spectrum. *Journal of the Atmospheric Sciences*, 66(2):450–467, 2009. doi: 10.1175/2008JAS2653.1.

Peter Jan van Leeuwen. Nonlinear data assimilation in geosciences: an extremely efficient particle filter. *Quarterly Journal of the Royal Meteorological Society*, 136(653):1991–1999, 2010. doi: 10.1002/qj.699.

Mark Veillette, Siddharth Samsi, and Chris Mattioli. SEVIR: A storm event imagery dataset for deep learning applications in radar and satellite meteorology. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22009–22019, 2020.

Pengpeng Xiao, Phillip Si, and Peng Chen. LD-EnSF: Synergizing latent dynamics with ensemble score filters for fast data assimilation with sparse observations. *arXiv preprint arXiv:2411.19305*, 2024. doi: 10.48550/arXiv.2411.19305.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3813–3824, 2023. doi: 10.1109/ICCV51070.2023.00355.

A Extended Related Work

Classical filtering and variational DA. The Kalman filter (Kalman, 1960) provides the closed-form filtering posterior for linear-Gaussian state-space models. For nonlinear dynamics, the ensemble Kalman filter (EnKF) (Evensen, 1994) replaces the analytic update with a Gaussian sample-covariance correction over an ensemble propagated through the true dynamics. Variants such as the ensemble transform Kalman filter (ETKF, Bishop et al., 2001) and its localized form (LETKF, Hunt et al., 2007) extend this idea to high-dimensional geophysical systems and are widely used in operational numerical weather prediction (Houtekamer et al., 2014a; Schraff et al., 2016). A complementary and dominant methodology, variational data assimilation, casts DA as deterministic optimization of a cost functional over the state trajectory; 3D-Var (Lorenc, 1986b) and 4D-Var (Courtier et al., 1994) are the canonical examples. Variational methods produce point estimates rather than posterior distributions, and are therefore complementary to the probabilistic filtering setting we consider in this work.

Particle filters and the curse of dimensionality. Particle filters (Gordon et al., 1993) represent the filtering posterior with a weighted ensemble and place no Gaussianity or linearity assumptions on the dynamics or observation model. The bootstrap PF (BPF) uses the transition prior as the proposal and reweights by the observation likelihood. In the more general sequential importance resampling (SIR) framework (Doucet et al., 2000), particles are drawn from an arbitrary proposal $q(x_t \mid x_{t-1}, o_t)$ and weighted by the SIR correction, recovering consistent posterior estimates as $N \to \infty$. The conditional variance of the incremental importance weight is minimized by the locally optimal proposal $q^\star(x_t \mid x_{t-1}, o_t) = p(x_t \mid x_{t-1}, o_t)$ (Doucet et al., 2000), but $q^\star$ is intractable outside special cases such as linear-Gaussian models. Without a good proposal, particle filters degenerate as the state dimension grows: the ensemble size required to keep weights from collapsing onto a few particles scales exponentially with the effective state dimension (Snyder et al., 2008b; Bengtsson et al., 2008; Snyder et al., 2015). Auxiliary particle filters (APF, Pitt & Shephard, 1999) provide a one-step observation look-ahead that yields modest improvements in moderate dimensions but does not avoid the exponential scaling. Approximations to $q^\star$ based on local linearization or Gaussian assumptions reintroduce the modeling restrictions that motivated moving away from the Kalman family in the first place.

Localized particle filters. For spatially extended systems, localization exploits sparse spatial dependencies in the posterior to break the global filtering problem into overlapping local subproblems whose effective dimension is set by a localization radius rather than by the full state dimension (Lauritzen, 1996; Hunt et al., 2007). Poterjoy (2016) introduces the localized particle filter (LPF) for high-dimensional geophysical systems by tapering observation likelihoods and resampling per site. Penny & Miyoshi (2016) and Farchi & Bocquet (2018) develop and compare several variants. These approaches all target the bootstrap PF, where the SIR weight reduces to the observation likelihood and decomposes naturally over space because the proposal is the unconditional transition prior. For a learned proposal, the proposal log-density does not factor unless the velocity network itself respects the locality structure, which motivates the patch-based design of L-FPPF. Our construction draws on the analysis of Gottwald et al. (2025), who show that for distributions with sparse conditional dependencies the score function (and equivalently the rectified-flow velocity) at a given site depends effectively only on a local neighborhood, with approximation error decaying exponentially in the window radius.

Learned proposals for sequential Monte Carlo. Several lines of work learn proposal distributions for SMC. Neural adaptive SMC (NASMC, Gu et al., 2015) parameterizes the proposal as a Gaussian or mixture density network and trains by particle-weighted log-likelihood, running an inner particle filter inside the training loop; this can be unstable when the proposal is far from the posterior. Variational SMC (Naesseth et al., 2018; Le et al., 2018; Maddison et al., 2017) similarly runs a differentiable PF during training and optimizes a variational lower bound on the marginal likelihood. InfNN (Paige & Wood, 2016) avoids the inner filter by fitting an autoregressive conditional density (RNADE) offline by maximum likelihood, but restricts the proposal to a tractable factorized form. Earlier adaptive-proposal work (Cornebise et al., 2008; Branchini & Elvira, 2020) optimizes within parametric families with closed-form density. To our knowledge, FPPF is the first to instantiate the proposal with a conditional generative model carrying no distributional assumption while retaining tractable density evaluation, in this case via the instantaneous change of variables formula for flow matching. Training reduces to velocity regression on offline simulated tuples (x_{t-1}, o_t, x_t), and is remarkably stable and efficient compared to our proposal learning baselines. A separate thread, differentiable particle filters (Karkus et al., 2018; Jonschkowski et al., 2018; Corenflos et al., 2021; Chen & Li, 2023; 2024), jointly learns multiple state-space components (transition, observation, proposal) end-to-end through a differentiable resampling operation. Differentiable PFs target the joint parameter and state estimation problem, whereas we assume known dynamics and observation models and learn only the proposal. Optimizing through the filter over long chaotic trajectories also remains difficult in practice, which is part of the motivation for our offline training scheme.

Generative-model approaches to data assimilation. A growing line of work uses score- and flow-based generative models for DA, falling into two categories. The first replaces or augments the predict step with a learned conditional generator. Score-based data assimilation (SDA, Rozet & Louppe, 2023) learns a score over short trajectory windows and conditions on observations via classifier guidance. DiffDA (Huang et al., 2024) adapts a pretrained weather forecaster, GraphCast (Lam et al., 2023), to the assimilation setting through diffusion-based inpainting. FlowDAS (Chen et al., 2025) learns a stochastic-interpolant transition model and applies guidance at observation times. Because these methods absorb the observation into the generator, the resulting iterates are not draws from the filtering posterior, and errors accumulate over long rollouts. The second category uses generative models for the analysis step and the distribution representation itself. The ensemble score filter (EnSF, Bao et al., 2024) is a training-free method that estimates the score of the filtering density at each assimilation step via a Monte Carlo estimator and produces the analysis ensemble by reverse-time SDE sampling, with observations folded in through likelihood-gradient guidance. Latent-EnSF (Si & Chen, 2025) extends this to a VAE-induced latent space to handle sparse observations, and LD-EnSF (Xiao et al., 2024) couples the latent representation with a learned surrogate of the dynamics. The ensemble flow filter (EnFF, Transue et al., 2025) is also training-free, but replaces reverse-time SDE sampling with flow matching using a Monte Carlo estimator of the marginal vector field and incorporates observations via guidance, giving faster sampling than EnSF. These methods differ from FPPF in two ways. First, they incorporate observations through guidance, an approximate gradient correction to the generative dynamics, rather than a Bayesian update; FPPF instead retains the SIR correction so that importance weights provide a principled update step and the empirical measure converges to the filtering posterior as $N \to \infty$. Second, they target the representation of the analysis distribution, whereas FPPF targets the proposal. The two are complementary: an FPPF-style learned proposal could in principle be combined with latent or score-based ensemble representations to benefit from both observation-informed propagation and compact distribution representation, which we view as a promising direction for future work. A concurrent method, DAISI (Andrae et al., 2026), pairs a stationary pretrained generative prior with an inverse-sampling step that folds forecast information into the latent before assimilating observations through guidance; like EnSF and EnFF it targets the analysis distribution via guidance.

Hybrid ML and DA methods. A complementary line of work learns surrogate dynamics from data and then plugs the surrogate into a classical filter. Brajard et al. (2020) train a neural network on EnKF analyses to emulate the dynamics of chaotic systems, and Bocquet et al. (2020) formulate joint Bayesian inference of the dynamical model and the state. AD-EnKF (Chen et al., 2022) performs auto-differentiable EnKF for joint state and parameter estimation. These methods target the transition model, whereas FPPF targets the proposal. The two threads are complementary: a learned dynamics model with a tractable transition density can be combined with a learned proposal in the

SIR framework. Extending FPPF to settings where the transition density itself must be approximated, for example by an emulator, is left for future work (Section 6).

A.1 Additional experiments

Convergence for linear-Gaussian setting. To verify that the rectified flow proposal converges to the correct target, we test on a synthetic linear Gaussian state-space model (full settings in Appendix D), where the locally optimal one-step proposal $q^\star(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t)$ admits a closed form. Using the same architecture as our L96 experiments, we train q_ϕ on a $d_x{=}8$ system for 200 epochs.

The left panel of Figure 6 reports the Wasserstein W_2 distance between q_ϕ and $q^\star$, averaged over 500 test set conditioning pairs $(\mathbf{x}_{t-1}, \mathbf{o}_t)$. It decreases over training to 0, indicating convergence to the closed-form optimal proposal. The dotted horizontal line indicates the W_2 distance between the bootstrap proposal $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ and the closed-form optimal proposal $q^\star(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t)$.

The middle panel reports a one-step ESS. For each conditioning pair we draw $N{=}250$ samples $\mathbf{x}_t^{(i)} \sim q_\phi(\cdot \mid \mathbf{x}_{t-1}, \mathbf{o}_t)$, form the one-step SIR importance weights

$$\tilde{w}^{(i)} \propto \frac{p(\mathbf{o}_t \mid \mathbf{x}_t^{(i)})\, p(\mathbf{x}_t^{(i)} \mid \mathbf{x}_{t-1})}{q_\phi(\mathbf{x}_t^{(i)} \mid \mathbf{x}_{t-1}, \mathbf{o}_t)},$$

and compute $\mathrm{ESS} = 1/\sum_i (\bar{w}^{(i)})^2$ from the normalized weights $\bar{w}^{(i)}$, averaging over the 500 pairs. This is the ESS the filter would achieve after a single observation update with the proposal. It attains its ceiling of $N{=}250$ when the incremental weights are uniform ($\bar{w}^{(i)} = 1/N$), which is exactly what the optimal proposal achieves, since under $q^\star$ the incremental weight is independent of the sampled $\mathbf{x}_t$. The bootstrap baseline collapses to near 80 (dashed line), while q_ϕ stabilizes near 250 over training.

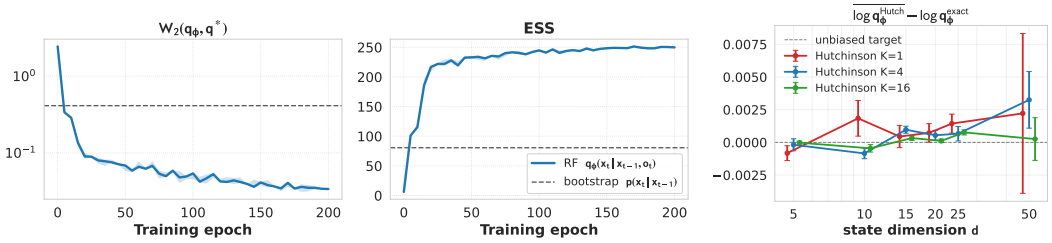


Figure 6: **Left:** Wasserstein W_2 distance from the learned proposal q_ϕ to the closed-form optimal proposal $q^\star$ over training epochs, averaged over 500 test conditioning pairs; it decreases toward 0, while the dashed line is the bootstrap proposal's W_2 to $q^\star$. **Middle:** one-step ESS out of $N{=}250$ over training epochs; q_ϕ rises toward the maximum of 250, while the bootstrap proposal's (dashed) is near 80. **Right:** bias of the Hutchinson trace estimator relative to the exact trace on the Lorenz-96 proposals across state dimension d_x, for $K \in \{1, 4, 16\}$ probes; points show mean bias where ideal is 0. Error bars indicate standard deviation across $N_{\text{pairs}}{=}10$ conditioning pairs.

Hutchinson trace estimator. Computing $\log q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t)$ requires the divergence of the proposal velocity field, which we estimate with the Hutchinson trace estimator (Section 3.2). One concern is that the single-probe estimator ($K{=}1$) injects noise into $\log q_\phi$ and could inflate importance weight variance. We quantify the single-probe estimator error on the trained Lorenz-96 RF proposals from Section 5.1. At $N_{\text{pairs}}{=}10$ test-set conditioning pairs we draw $\mathbf{x}_t \sim q_\phi$, evaluate the Hutchinson estimate of $\log q_\phi$ at $K \in \{1, 4, 16\}$ probes ($M{=}200$ random-probe seeds each), and measure its bias and standard deviation against the deterministic exact-trace value. The empirical bias is consistent with zero at every d_x within pair-to-pair, see Figure 6 (right). Estimator standard deviation stays below 0.05 for $d_x \leq 25$ but grows to 0.247 at $d_x{=}50$ for $K{=}1$; it is substantially reduced for $K = 16$. While we keep $K{=}1$ by default for efficiency, at high dimensions increasing K at the expense of compute may improve performance.

B Algorithm

The FPPF algorithm is given in 1.

Algorithm 1 Flow Proposal Particle Filter (FPPF)

Require: Trained velocity field v_{ϕ}, observations $\mathbf{o}_{1:T}$, particles N, threshold N_{eff}

Initialize: $\mathbf{x}_0^{(i)} \sim p_0(\mathbf{x})$, $w_0^{(i)} = 1/N$ for $i = 1, \dots, N$
for $t = 1, \dots, T$ **do**
  for $i = 1, \dots, N$ **do**
    [PREDICT] Sample $\mathbf{x}_t^{(i)} \sim q_{\phi}(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{o}_t)$ using Eq. (3)
    [UPDATE] Compute $\widetilde{w}_t^{(i)}$ using Eq. (4)
  end for
  Normalize $w_t^{(i)} \leftarrow \tilde{w}_t^{(i)} / \sum_{j=1}^{N} \tilde{w}_t^{(j)}$; compute $\text{ESS}_t \leftarrow 1/\sum_{i=1}^{N} (w_t^{(i)})^2$
  if $\text{ESS}_t < N_{\text{eff}}$ **then [RESAMPLE]** draw $\{\mathbf{x}_t^{(i)}\}$ according to $\{w_t^{(i)}\}$ and reset $w_t^{(i)} \leftarrow 1/N$
end for
Return weighted particles $\{(\mathbf{x}_t^{(i)}, w_t^{(i)})\}_{t=1}^{T}$

C Experimental Setup

C.1 Data split

For Lorenz systems we generate 2048 simulated trajectories and split 80/10/10 into train/validation/test. A subset of the test split is used to report results due to compute constraints; exact subset sizes are listed per experiment.

C.2 FPPF architecture and hyperparameters

Global backbones. For global comparisons, FPPF, FlowDAS, DAISI, FA-APF, NASMC, and InfNN use matched feature backbones, with method-specific output heads. **(i) For L96**, the backbone is a 1D ResNet with circular padding, kernel size 5, SiLU activations, and adaptive layer normalization (AdaLN) for flow-time conditioning. Each residual block contains two convolutions, and the conditioning and output projections are zero-initialized. **(ii) For L63**, FPPF uses a flat MLP with SiLU activations and concatenation conditioning on the previous state and observation. **(ii) For L63**, For SQG, we use the released DAISI two-channel U-Net, with base width 32, channel multipliers $(2, 2, 2)$, Fourier time embeddings, and attention at 32×32 resolution.

L-FPPF backbones. **(i) For L96**, the localized velocity field is a 1D ResNet operating on periodic patches of radius $r = 4$ and width $W = 9$. It uses 64 channels, four residual blocks with kernel size 3, and a 64-dimensional flow-time embedding supplied through AdaLN. The four input channels contain the evolving flow state, previous state, observation values, and observation mask. **(ii) For SQG**, a periodic convolutional encoder with 64 channels and dilations $(1, 2, 4, 8)$ processes the forecast and observation conditioning. The velocity network uses a three-block, width-128 pointwise head, a 64-dimensional time embedding, and a 3×3 neighborhood of the evolving two-channel state. **(iii) For SEVIR**, a U-Net predicts the forecast mean and context from the previous radar frame. The residual velocity network uses 48 context channels, a width-128 head with two residual blocks, and a 32-dimensional time embedding. Its 3×3 spatial interactions are restricted to disjoint 16×16 blocks. A multiscale infrared encoder supplies observation conditioning through a trainable velocity correction, while the transition backbone remains frozen.

Conditioning. We apply two training-only conditioning regularizers. (1) Similar to classifier-free conditioning dropout with probability $p_{\text{cond}} = 0.1$ the conditioning $\mathbf{o}_t$ is zeroed before being fed to the backbone, so the network sees both conditional and unconditional batches with respect to the observation. (2) Previous-state corruption schedule: during training only, with annealed probability $p(t) = \max(p_{\text{min}}, p_0\,(1 - t/T))$ we corrupt $\mathbf{x}_{t-1}$ by zeroing a random subset of its coordinates of size $\lceil r_{\text{mask}}\, d_x \rceil$ (no additive Gaussian noise). We use $p_0 = 0.3$, $p_{\text{min}} = 0.05$, $r_{\text{mask}} = 0.4$, with T set to the total number of training steps. This forces the conditional network to use $\mathbf{o}_t$ rather than over-relying on the clean $\mathbf{x}_{t-1}$ on systems whose forward dynamics is highly predictable. The corruption is disabled at inference.

Training All models are trained for a minimum of 80 epochs, with an early-stopping patience set to 50 epochs.

L-FPPF patch network. The localized velocity field u_ϕ is a small 1D ResNet operating on spatial patches of size $W = 2r + 1$. We use radius $r = 4$ (so $W = 9$), $C = 64$ channels, 2 residual blocks of kernel size 3, and a 64-dim flow-time embedding fed via AdaLN. Inputs are stacked as a 4-channel window $[\mathbf{z}_{[j-r:j+r]},\ \mathbf{x}_{t-1,[j-r:j+r]},\ \mathbf{o}^{\text{full}}_{t,[j-r:j+r]},\ m_{t,[j-r:j+r]}]$.

C.3 Baseline Implementation Details

Auxiliary particle filter (APF). We use the classical Pitt–Shephard (Pitt & Shephard, 1999) auxiliary-particle filter with the prior proposal $q = p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ and a point look-ahead. Concretely, the auxiliary adjustment multiplier is $m_t^{(i)} = g\Big(\mathbf{o}_t \,\Big|\, \boldsymbol{\mu}_t^{(i)}\Big)$, where $\boldsymbol{\mu}_t^{(i)} = \mathbb{E}\Big[\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)}\Big]$ is the deterministic transition mean (one Δt Euler step of the true dynamics, no process noise). With $q = p$, the APF incremental weight collapses to $\log w_t^{(i)} = \log g(\mathbf{o}_t \mid \mathbf{x}_t^{(i)}) - \log m_t^{(k_i)}$, where k_i is the ancestor selected in the first-stage resample drawn from $\lambda^{(i)} \propto w_{t-1}^{(i)} m_t^{(i)}$. Weights are reset to the second-stage incremental weight after each step, and a final ESS $< 0.33\,N$-triggered systematic resample is applied to keep the cloud healthy.

Ensemble Kalman filter (EnKF). We use EnKF with multiplicative covariance inflation. The inflation factor is tuned over $\lambda \in \{0.8, 0.9, 1.0, 1.1\}$ on the validation set and reported at the best-performing λ (lowest RMSE), ensemble size $N = 50$.

LETKF. Our LETKF (Hunt et al., 2007) uses a per-coordinate local analysis on a 1-D periodic spatial domain. For each state index j, observations are tapered by a Gaspari–Cohn polynomial $\rho(d/r)$ with periodic distance d (modulo d_x) and localization radius r, the local symmetric square-root analysis is solved in ensemble space, and the resulting analysis ensemble member $\mathbf{x}_j^{(i)}$ is assembled by stacking centre-coordinate analyses We tune inflation factor on a held-out validation subset over $\lambda \in \{0.8, 0.9, \ldots, 1.8\}$ and set r to match other localized baselines. For the L96-25 in-distribution and high-dim ($d_x \in \{50, 100, 500, 1000\}$) cells we use $r = 4$ (matched to the L-FPPF localization radius of $r = 4$ patches), $\lambda = 1.0$, ensemble size $N = 50$.

LBPF. The localized BPF is the block-localized particle filter of Farchi & Bocquet (2018): the spatial domain is partitioned into contiguous blocks and the analysis weights, ancestor resampling, and patchwork particle reassembly are computed independently within each block on tapered local likelihoods. We use block size 1 (per-coordinate analysis), with localization radius $r = 4$ (matched to L-FPPF), and systematic local resampling with the identity-preserving (adjustment-minimizing) ancestor-permutation tie-break of Farchi & Bocquet (2018). No post-regularization jitter is applied.

FlowDAS. For all FlowDAS experiments (Chen et al., 2025) we use the official implementation. The guidance scale is set to 0.01 (selected via ablation), $J = 21$ Monte Carlo sampling times, learning rate 3×10^{-4} with early stopping. Inference uses 50 first-order Euler steps. FlowDAS does not produce an ensemble by default; for CRPS we draw 20 independent rollouts per evaluation trajectory and compute CRPS from this empirical predictive distribution.

DAISI. For DAISI (Andrae et al., 2026), we train forecast priors according to the official implementation. Lorenz priors are trained with Adam at learning rate 10^{-4}, using batch size 512 for 60 epochs on L96 and batch size 256 for 20 epochs on L63. Sampler settings are selected on validation trajectories. L63 uses 100-step MMPS sampling. SQG uses 100 inverse and 100 forward steps, with MMPS guidance.

Learned FA-APF. For learned FA-APF (Savary et al., 2026), we train forecast priors according to the official implementation. Conditional transition models are trained with AdamW at learning rate 2×10^{-4} and batch size 128, for 60 epochs on L96 and SQG and 10 epochs on L63. Inference uses moment-matching guidance and auxiliary ancestor resampling, with 32 DDIM steps on L96 and 64 on observed L63 steps; unobserved L63 forecasts use 128 steps. All DDIM samplers use $\eta = 0.5$.

NASMC. We follow Gu et al. (2015) and report two variants: $\texttt{NASMC}_{K=1}$, the single-Gaussian baseline of Gu et al., and $\texttt{NASMC}_{K=3}$, the "-MD-" mixture-density variant with $K = 3$ components (paper default). We do *not* use the paper's LSTM history conditioning; conditioning is Markovian on $(\mathbf{x}_{t-1}, \mathbf{o}_t)$, matching the FPPF / InfNN comparison. At evaluation, the trained q_ϕ is plugged into our standard BPF and inherits all per-cell PF settings (N, ESS threshold, σ_x, σ_y, batch size) from the corresponding FPPF cell.

InfNN. We use `InfNN-RNADE` variant from the paper: coordinates are factorized in their natural index order $\prod_{j=1}^{d_x} q_\phi(x_t^{(j)} \mid x_t^{(<j)}, \mathbf{x}_{t-1}, \mathbf{o}_t)$, with each factor a $K = 3$-component diagonal Gaussian mixture produced by a 2-layer MLP of hidden size 128 (RNADE head, mirroring the original paper). At evaluation, the trained q_ϕ is plugged into our standard BPF and inherits all per-cell PF settings (N, ESS threshold, σ_x, σ_y, batch size) from the corresponding FPPF cell.

C.4 Inference time proposal cost

For all experiments we use $N_s = N_\ell = 32$ Euler integration steps for both forward sampling and reverse likelihood evaluation, on a uniform time grid for sampling and a γ=2 non-uniform power-back grid for likelihood, where $s_k = 1 - (1 - u_k)^\gamma$, $\gamma > 1$. This results in smaller step sizes closer to the target distribution. Empirically, we find that the non-uniform grid allowed us to decrease the number of integration steps from 100 to 32 with minimal performance degradation, compared to a uniform grid. A natural direction to reduce the inference-time cost of FPPF, which currently runs ∼32-step Euler integration twice per particle per step (once for sampling, once for likelihood evaluation), is to replace the rectified-flow teacher with a few-step student. Recent work on consistency models (Song et al., 2023), MeanFlow (Geng et al., 2025), and Shortcut models (Frans et al., 2025) produces high-quality samples in one or a handful of NFEs, and the F2D2 framework (Ai et al., 2026) extends this idea to fast *divergence* estimation, opening a path to also accelerate the likelihood evaluation step rather than only sampling. In our preliminary experiments, however, we found that the resulting few-step proposals were of substantially lower quality than the teacher: students tended to collapse toward the conditional mean, eroding the observation-informed structure that makes the proposal useful in the first place, and the residual bias in the distilled divergence degraded ESS over long horizons. We therefore report the 32-step teacher throughout the main paper. Bridging the gap between distillation quality and proposal informativeness is an interesting direction for reducing the compute cost of FPPF.

D Linear-Gaussian System Details

System. We use d_x=8 with $\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{w}_t$, $\mathbf{o}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t$, where $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$, $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$, and $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Writing $\mathbf{S}$ for the cyclic shift and $\mathbf{C}(\alpha)_{ij} = \alpha^{|i-j|}$ for a Toeplitz correlation, we set $\mathbf{A} = 0.92\,\mathbf{I} + 0.05\,\mathbf{S} + 0.02\,\mathbf{S}^\top$, $\mathbf{H} = \mathbf{I} + 0.25\,\mathbf{S} - 0.15\,\mathbf{S}^\top$, $\mathbf{Q} = 0.35^2(0.7\,\mathbf{I} + 0.3\,\mathbf{C}(0.5))$, $\mathbf{R} = 0.25^2(0.6\,\mathbf{I} + 0.4\,\mathbf{C}(0.7))$. $\mathbf{A}$ is stable ($\rho(\mathbf{A})$<1), $\mathbf{H}$ mixes neighbours asymmetrically, and both noises are spatially correlated. All states are observed.

Dataset. N_{traj}=1024 trajectories of length T=200 after a 100-step burn-in, split 80/10/10 by trajectory into train / val / test.

Optimal proposal. For this system $q^\star(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{o}_t)$ is Gaussian with

$$\mathbf{K} = \mathbf{Q}\mathbf{H}^\top(\mathbf{H}\mathbf{Q}\mathbf{H}^\top + \mathbf{R})^{-1}, \quad \boldsymbol{\mu}^\star = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{K}(\mathbf{o}_t - \mathbf{H}\mathbf{A}\mathbf{x}_{t-1}), \quad \boldsymbol{\Sigma}^\star = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{Q}.$$

$\boldsymbol{\Sigma}^\star$ is independent of $(\mathbf{x}_{t-1}, \mathbf{o}_t)$, so all conditioning enters through $\boldsymbol{\mu}^\star$. This is the target the learned proposal q_ϕ must match.

Bootstrap reference. The dynamics prior $q_b(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{A}\mathbf{x}_{t-1}, \mathbf{Q})$ ignores $\mathbf{o}_t$ and serves as the floor a useful learned proposal must beat.

E Full Lorenz-96 Results

Complete results are in Table 1 and Table 2.

Method	State Dimension d_x = 5	10	15	20	25	50
RMSE ↓						
BPF	0.303 ± 0.030	0.357 ± 0.036	0.364 ± 0.029	0.419 ± 0.038	0.480 ± 0.046	0.743 ± 0.122
APF	0.304 ± 0.036	0.353 ± 0.033	0.356 ± 0.029	0.401 ± 0.037	0.450 ± 0.039	0.673 ± 0.099
EnKF	0.287 ± 0.050	0.326 ± 0.056	0.365 ± 0.056	0.416 ± 0.069	0.448 ± 0.074	0.724 ± 0.151
NASMC-Gauss	0.430 ± 0.068	0.529 ± 0.084	0.573 ± 0.101	0.617 ± 0.146	0.653 ± 0.145	0.762 ± 0.167
NASMC-MD	0.471 ± 0.071	0.533 ± 0.100	0.577 ± 0.108	0.614 ± 0.102	0.659 ± 0.128	0.751 ± 0.143
InfNN	0.524 ± 0.192	1.000 ± 0.411	1.363 ± 0.316	2.117 ± 0.415	2.467 ± 0.392	3.396 ± 0.231
FlowDAS	0.438 ± 0.191	0.562 ± 0.366	0.505 ± 0.293	0.735 ± 0.282	0.603 ± 0.295	0.970 ± 0.313
DAISI	0.207 ± 0.028	**0.259 ± 0.046**	0.276 ± 0.055	0.287 ± 0.052	0.278 ± 0.040	**0.297 ± 0.045**
FA-APF	0.266 ± 0.028	0.336 ± 0.038	0.396 ± 0.039	0.477 ± 0.065	0.524 ± 0.063	0.782 ± 0.124
FPPF	**0.189 ± 0.027**	0.287 ± 0.070	**0.217 ± 0.043**	**0.238 ± 0.037**	**0.235 ± 0.032**	0.323 ± 0.114
CRPS ↓						
BPF	0.195 ± 0.013	0.212 ± 0.018	0.212 ± 0.012	0.233 ± 0.018	0.262 ± 0.023	0.439 ± 0.078
APF	0.195 ± 0.015	0.210 ± 0.016	0.209 ± 0.012	0.225 ± 0.016	0.246 ± 0.018	0.381 ± 0.063
EnKF	0.174 ± 0.024	0.186 ± 0.026	0.203 ± 0.029	0.229 ± 0.037	0.246 ± 0.041	0.422 ± 0.094
NASMC-Gauss	0.263 ± 0.041	0.332 ± 0.058	0.358 ± 0.067	0.393 ± 0.092	0.423 ± 0.095	0.501 ± 0.097
NASMC-MD	0.292 ± 0.044	0.336 ± 0.067	0.360 ± 0.069	0.391 ± 0.067	0.423 ± 0.081	0.493 ± 0.083
InfNN	0.325 ± 0.142	0.662 ± 0.301	0.908 ± 0.224	1.448 ± 0.283	1.711 ± 0.264	2.450 ± 0.176
FlowDAS	0.281 ± 0.109	0.330 ± 0.213	0.277 ± 0.162	0.411 ± 0.163	0.316 ± 0.152	0.530 ± 0.177
DAISI	0.191 ± 0.009	0.212 ± 0.022	0.218 ± 0.019	0.221 ± 0.021	0.219 ± 0.016	0.221 ± 0.015
FA-APF	0.173 ± 0.012	0.199 ± 0.017	0.221 ± 0.016	0.259 ± 0.030	0.280 ± 0.027	0.397 ± 0.063
FPPF	**0.115 ± 0.013**	**0.165 ± 0.044**	**0.122 ± 0.022**	**0.132 ± 0.018**	**0.132 ± 0.017**	**0.183 ± 0.056**

Table 1: Lorenz-96 results on the arctan observation operator at observation noise $\sigma_y = 0.2$ across state dimensions. Best results in **bold**; second best underlined.

F Full High-dimensional Lorenz-96 Results

Complete results are in Table 3 and Table 4.

Method	State Dimension d_x					
	5	10	15	20	25	50
RMSE ↓						
BPF	0.279 ± 0.034	0.411 ± 0.070	0.404 ± 0.051	0.591 ± 0.318	0.803 ± 0.474	1.942 ± 0.552
APF	0.288 ± 0.032	0.416 ± 0.061	0.429 ± 0.071	0.606 ± 0.367	0.772 ± 0.471	1.878 ± 0.560
EnKF	0.757 ± 0.976	1.661 ± 1.580	2.621 ± 1.543	3.157 ± 1.319	3.474 ± 1.149	4.353 ± 0.531
NASMC-Gauss	0.556 ± 0.332	0.957 ± 0.690	1.060 ± 0.650	1.248 ± 0.699	1.502 ± 0.707	1.425 ± 0.486
NASMC-MD	0.639 ± 0.419	0.857 ± 0.616	1.050 ± 0.574	1.416 ± 0.796	1.184 ± 0.628	1.605 ± 0.514
InfNN	0.962 ± 1.086	3.282 ± 0.914	3.599 ± 0.624	4.213 ± 0.425	4.314 ± 0.424	4.726 ± 0.262
FlowDAS	0.354 ± 0.132	0.560 ± 0.387	0.510 ± 0.283	0.740 ± 0.288	0.629 ± 0.320	0.945 ± 0.321
DAISI	0.694 ± 0.158	1.131 ± 0.315	1.258 ± 0.254	1.257 ± 0.248	1.283 ± 0.249	1.304 ± 0.182
FA-APF	0.227 ± 0.023	0.359 ± 0.057	0.457 ± 0.083	0.639 ± 0.214	0.851 ± 0.381	2.403 ± 0.669
FPPF	**0.201 ± 0.032**	**0.321 ± 0.125**	**0.237 ± 0.059**	**0.382 ± 0.310**	**0.254 ± 0.068**	**0.519 ± 0.298**
CRPS ↓						
BPF	0.205 ± 0.013	0.250 ± 0.036	0.242 ± 0.023	0.332 ± 0.176	0.456 ± 0.307	1.213 ± 0.396
APF	0.207 ± 0.013	0.250 ± 0.030	0.253 ± 0.033	0.342 ± 0.226	0.431 ± 0.302	1.143 ± 0.385
EnKF	0.520 ± 0.772	1.145 ± 1.203	1.761 ± 1.157	2.102 ± 1.022	2.368 ± 0.941	3.102 ± 0.491
NASMC-Gauss	0.358 ± 0.263	0.657 ± 0.539	0.703 ± 0.456	0.823 ± 0.482	0.961 ± 0.487	0.901 ± 0.311
NASMC-MD	0.423 ± 0.335	0.581 ± 0.472	0.697 ± 0.410	0.920 ± 0.542	0.773 ± 0.421	0.995 ± 0.333
InfNN	0.684 ± 0.915	2.441 ± 0.747	2.606 ± 0.538	3.149 ± 0.381	3.227 ± 0.401	3.651 ± 0.244
FlowDAS	0.233 ± 0.075	0.331 ± 0.229	0.283 ± 0.154	0.411 ± 0.163	0.330 ± 0.165	0.511 ± 0.181
DAISI	0.411 ± 0.054	0.608 ± 0.153	0.663 ± 0.117	0.662 ± 0.118	0.670 ± 0.119	0.670 ± 0.086
FA-APF	0.174 ± 0.009	0.220 ± 0.026	0.260 ± 0.040	0.343 ± 0.105	0.446 ± 0.187	1.345 ± 0.450
FPPF	**0.125 ± 0.017**	**0.191 ± 0.083**	**0.134 ± 0.032**	**0.223 ± 0.205**	**0.142 ± 0.037**	**0.289 ± 0.143**

Table 2: Lorenz-96 results on the quadratic capped observation operator at observation noise $\sigma_y = 0.2$ across state dimensions. Best results in **bold**; second best underlined.

Method	State Dimension d_x			
	50	100	500	1000
RMSE ↓				
BPF	0.743 ± 0.122	1.232 ± 0.224	2.421 ± 0.146	2.776 ± 0.090
APF	0.669 ± 0.087	1.082 ± 0.183	2.386 ± 0.135	2.732 ± 0.089
EnKF	0.706 ± 0.133	1.187 ± 0.220	2.278 ± 0.162	2.522 ± 0.109
LBPF	0.366 ± 0.016	0.370 ± 0.011	0.372 ± 0.005	0.372 ± 0.004
LETKF	0.316 ± 0.022	0.319 ± 0.015	0.319 ± 0.007	0.318 ± 0.004
DAISI	0.296 ± 0.043	**0.295 ± 0.025**	0.303 ± 0.014	0.304 ± 0.010
FA-APF	0.966 ± 0.192	2.090 ± 0.437	2.695 ± 0.096	2.962 ± 0.084
L-FPPF	**0.287 ± 0.012**	0.300 ± 0.014	**0.293 ± 0.007**	**0.299 ± 0.004**
CRPS ↓				
BPF	0.439 ± 0.078	0.827 ± 0.150	1.757 ± 0.110	2.058 ± 0.070
APF	0.377 ± 0.053	0.707 ± 0.120	1.727 ± 0.103	2.021 ± 0.072
EnKF	0.411 ± 0.082	0.755 ± 0.144	1.599 ± 0.116	1.803 ± 0.084
LBPF	0.212 ± 0.007	0.213 ± 0.004	0.213 ± 0.002	0.213 ± 0.002
LETKF	0.177 ± 0.009	0.178 ± 0.006	0.177 ± 0.003	0.177 ± 0.002
DAISI	0.222 ± 0.015	0.222 ± 0.008	0.224 ± 0.004	0.224 ± 0.003
FA-APF	0.508 ± 0.109	1.135 ± 0.194	1.582 ± 0.057	1.729 ± 0.051
L-FPPF	**0.156 ± 0.006**	**0.162 ± 0.006**	**0.158 ± 0.003**	**0.161 ± 0.002**

Table 3: High-dimensional Lorenz-96 results with local filtering methods on the observation operator $h(\mathbf{x}) = \arctan(\mathbf{x})$. Best results in **bold**; second best underlined.

Method	State Dimension d_x			
	50	100	500	1000
RMSE ↓				
BPF	1.942 ± 0.552	2.534 ± 0.359	3.081 ± 0.124	3.245 ± 0.090
APF	1.908 ± 0.540	2.457 ± 0.356	3.087 ± 0.124	3.236 ± 0.087
EnKF	4.317 ± 0.538	4.623 ± 0.360	3.628 ± 0.222	3.318 ± 0.139
LBPF	0.684 ± 0.226	0.698 ± 0.254	0.785 ± 0.132	0.812 ± 0.085
LETKF	1.935 ± 0.785	2.017 ± 0.640	2.152 ± 0.249	2.144 ± 0.175
DAISI	1.266 ± 0.188	1.269 ± 0.108	1.271 ± 0.057	1.275 ± 0.038
FA-APF	3.055 ± 0.615	5.131 ± 0.590	5.329 ± 0.180	5.634 ± 0.133
L-FPPF	$\mathbf{0.670 \pm 0.339}$	$\mathbf{0.669 \pm 0.245}$	$\mathbf{0.698 \pm 0.125}$	$\mathbf{0.752 \pm 0.075}$
CRPS ↓				
BPF	1.213 ± 0.396	1.708 ± 0.278	2.268 ± 0.100	2.429 ± 0.072
APF	1.168 ± 0.369	1.633 ± 0.277	2.267 ± 0.106	2.421 ± 0.075
EnKF	3.065 ± 0.500	3.392 ± 0.337	2.629 ± 0.190	2.410 ± 0.117
LBPF	0.366 ± 0.091	0.366 ± 0.106	0.375 ± 0.041	0.383 ± 0.027
LETKF	0.954 ± 0.439	0.951 ± 0.346	0.979 ± 0.159	0.970 ± 0.115
DAISI	0.655 ± 0.088	0.652 ± 0.051	0.651 ± 0.026	0.654 ± 0.018
FA-APF	1.865 ± 0.461	3.481 ± 0.423	3.816 ± 0.143	4.085 ± 0.103
L-FPPF	$\mathbf{0.358 \pm 0.158}$	$\mathbf{0.338 \pm 0.090}$	$\mathbf{0.337 \pm 0.043}$	$\mathbf{0.349 \pm 0.024}$

Table 4: High-dimensional Lorenz-96 results with local filtering methods on the observation operator $h(\mathbf{x}) = \min(\mathbf{x}^4, 10)$. Best results in **bold**; second best underlined.