SIAM J. MATH. DATA SCI.
Vol. 4, No. 2, pp. 801–833

Autodifferentiable Ensemble Kalman Filters*

Yuming Chen†, Daniel Sanz-Alonso†, and Rebecca Willett†

Abstract. Data assimilation is concerned with sequentially estimating a temporally evolving state. This task, which arises in a wide range of scientific and engineering applications, is particularly challenging when the state is high-dimensional and the state-space dynamics are unknown. This paper introduces a machine learning framework for learning dynamical systems in data assimilation. Our autodifferentiable ensemble Kalman filters (AD-EnKFs) blend ensemble Kalman filters for state recovery with machine learning tools for learning the dynamics. In doing so, AD-EnKFs leverage the ability of ensemble Kalman filters to scale to high-dimensional states and the power of automatic differentiation to train high-dimensional surrogate models for the dynamics. Numerical results using the Lorenz-96 model show that AD-EnKFs outperform existing methods that use expectation-maximization or particle filters to merge data assimilation and machine learning. In addition, AD-EnKFs are easy to implement and require minimal tuning.

Key words. ensemble Kalman filters, autodifferentiation, data assimilation, machine learning

AMS subject classifications. 62M05, 68T09, 68T07, 86-08

DOI. 10.1137/21M1434477

1. Introduction.

Time series of data arising across geophysical sciences, remote sensing, automatic control, and a variety of other scientific and engineering applications often reflect observations of an underlying dynamical system operating in a latent state-space. Estimating the evolution of this latent state from data is the central challenge of data assimilation (DA) [44, 30, 82, 54, 75]. However, in these and other applications, we often lack an accurate model of the underlying dynamics, and the dynamical model needs to be learned from the observations to perform DA. This paper introduces autodifferentiable ensemble Kalman filters (AD-EnKFs), a machine learning (ML) framework for the principled co-learning of states and dynamics. This framework enables learning in three core categories of unknown dynamics: (a) parametric dynamical models with unknown parameter values; (b) fully unknown dynamics captured using neural network (NN) surrogate models; and (c) inaccurate or partially known dynamical models that can be improved using NN corrections. AD-EnKFs are designed to scale to high-dimensional states, observations, and NN surrogate models.

*Received by the editors July 19, 2021; accepted for publication (in revised form) March 26, 2022; published electronically June 23, 2022.

https://doi.org/10.1137/21M1434477

Funding: The first and second authors were supported by NSF DMS-2027056. The first and third authors were supported by NSF OAC-1934637. The second and third authors were supported by DOE DE-SC0022232. The second author received support from the FBBVA through a start-up grant. The third author was supported by AFOSR FA9550-18-1-0166, DOD FA9550-18-1-0166, DOE DE-AC02-06CH11357, NSF DMS-1925101, NSF DMS-1930049, and NSF DMS-2023109.

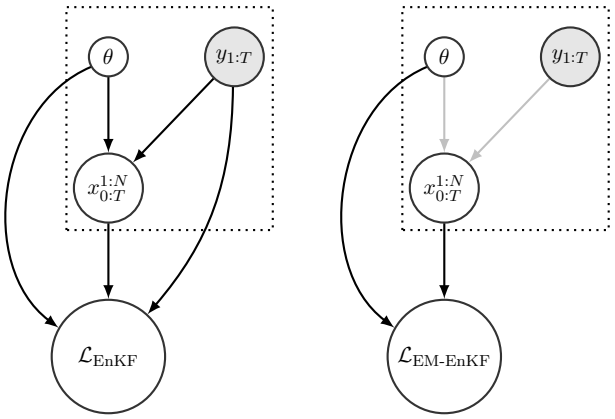
†University of Chicago, Chicago, IL 60637 USA (ymchen@uchicago.edu, sanzalonso@uchicago.edu, willett@g.uchicago.edu).

In order to describe the main idea behind the AD-EnKF framework, let us introduce briefly the problem of interest. Our setting will be formalized in section 2 below. Let $x_{0:T} := \{x_t\}_{t=0}^T$ be a time-homogeneous *state process* with transition kernel $p_\theta(x_t|x_{t-1})$ parameterized by a vector θ. For instance, θ may contain unknown parameters of a parametric dynamical model or the parameters of an NN surrogate model for the dynamics. Our aim is to learn θ from partial and noisy observations $y_{1:T} := \{y_t\}_{t=1}^T$ of the state, and thereby learn the unknown dynamics and estimate the state process. The AD-EnKF framework learns θ iteratively. Each iteration consists of three steps: (i) use EnKF to compute an estimate $\mathcal{L}_{\text{EnKF}}(\theta)$ of the data log-likelihood $\mathcal{L}(\theta) := \log p_\theta(y_{1:T})$; (ii) use autodifferentiation ("autodiff") to compute the gradient $\nabla_\theta \mathcal{L}_{\text{EnKF}}(\theta)$; and (ii′) take a gradient ascent step. Filtered estimates of the state are obtained using the learned dynamics.

The EnKF, reviewed in section 3, estimates the data log-likelihood using an ensemble of particles. Precisely, given a transition kernel $p_\theta(x_t|x_{t-1})$, the EnKF generates particles $x_{0:T}^{1:N} := \{x_t^n\}_{\substack{t=0,\dots T \\ n=1,\dots,N}}$; here x_t^n represents a generic particle that approximates the state x_t at discrete time $t \in \{0,\dots,T\}$, and N denotes the ensemble size. The log-likelihood estimate $\mathcal{L}_{\text{EnKF}}(\theta)$ depends on θ through these particles and also through the given transition kernel. Differentiating the map $\theta \mapsto \mathcal{L}_{\text{EnKF}}(\theta)$ in step (ii) of AD-EnKF involves differentiating *both* the map $\theta \mapsto x_{0:T}^{1:N}$ from parameter to EnKF particles and the map $(\theta, x_{0:T}^{1:N}) \mapsto \mathcal{L}_{\text{EnKF}}(\theta)$ from parameters and EnKF particles to EnKF log-likelihood estimate. *A key feature of our approach is that $\theta \mapsto \mathcal{L}_{EnKF}(\theta)$ can be autodifferentiated using the reparameterization trick ([48] and section 4.1) and autodiff capabilities of NN software libraries* such as PyTorch [70], JAX [11], and Tensorflow [1]. Automatic differentiation is different from numerical differentiation in that derivatives are computed exactly through compositions of elementary functions whose derivatives are known, as opposed to finite difference approximations that cause discretization errors.

The AD-EnKF framework represents a significant conceptual and methodological departure from existing approaches to blend DA and ML based on the expectation-maximization (EM) framework; see Figure 1. Specifically, at each iteration, EM methods that build on the EnKF [71, 12, 9] employ a surrogate likelihood $\mathcal{L}_{\text{EM-EnKF}}(\theta\,; x_{0:T}^{1:N})$ where the particles $x_{0:T}^{1:N}$ are generated by EnKF and *fixed*. Importantly, EM methods compute gradients used to learn dynamics by differentiating only through the θ-dependence in $\mathcal{L}_{\text{EM-EnKF}}$ that does not involve the particles. In particular, in contrast to AD-EnKF, the map $\theta \mapsto x_{0:T}^{1:N}$ from parameter to EnKF particles is *not* differentiated. Moreover, the performance of EM methods is sensitive to the specific choice of EnKF algorithm in use, and the tuning of algorithmic parameters of EM can be challenging [12, 9]. Our numerical experiments suggest that, even when optimally tuned, EM methods underperform AD-EnKF in high-dimensional regimes. The better performance of AD-EnKF may be explained by the additional gradient information obtained by differentiating the map $\theta \mapsto x_{0:T}^{1:N}$.

The AD-EnKF framework also represents a methodological shift from existing differentiable particle filters (PFs) [65, 61, 56]. Similar to AD-EnKF, these methods rely on autodiff of a map $\theta \mapsto \mathcal{L}_{\text{PF}}(\theta)$, where the log-likelihood estimate $\mathcal{L}_{\text{PF}}(\theta)$ depends on θ through weighted particles $(w_{0:T}^{1:N}, x_{0:T}^{1:N})$ obtained by running a PF with transition kernel $p_\theta(x_t|x_{t-1})$. However, the use of PF suffers from two caveats. First, it is not possible to autodifferentiate directly



(a) AD-EnKF (b) EM-EnKF

With AD-EnKF, parameters θ and observations $y_{1:T}$ are used to generate EnKF particles $x_{0:T}^{1:N}$; the particles together with θ and $y_{1:T}$ are used to compute the likelihood $\mathcal{L}_{\text{EnKF}}$, and the gradient $\nabla_\theta \mathcal{L}_{\text{EnKF}}$ explicitly accounts for the map from θ to the particles $x_{0:T}^{1:N}$. In contrast, with EM-EnKF, the likelihood $\mathcal{L}_{\text{EM-EnKF}}$ is a function of θ and *fixed* particles $x_{0:T}^{1:N}$ generated by EnKF, so that computing the gradient $\nabla_\theta \mathcal{L}_{\text{EM-EnKF}}$ does *not* account for the map from θ to the particles $x_{0:T}^{1:N}$.

Figure 1. *Computational graph of AD-EnKF and EM-EnKF. Dashed squares represent computations performed by the EnKF. Gray arrows in* (b) *indicate that the construction of $\mathcal{L}_{EM\text{-}EnKF}$ is performed in two steps:* (1) *obtain $x_{0:T}^{1:N}$ from θ and $y_{1:T}$ (gray arrows), and* (2) *use θ and $x_{0:T}^{1:N}$ (no longer seen as a function of θ) to define $\mathcal{L}_{EM\text{-}EnKF}$. In contrast, those lines are black in* (a)*, indicating that in AD-EnKF the particles $x_{0:T}^{1:N}$ in $\mathcal{L}_{EnKF}$ are seen as varying with θ.*

through the PF resampling steps [65, 61, 56]. Second, while the PF log-likelihood estimates are consistent, their variance can be large, especially in high-dimensional systems. Moreover, their *gradient*, which is the quantity used to perform gradient ascent to learn θ, is not consistent [18].

1.1. Contributions. This paper seeks to set the foundations and illustrate the capabilities of the AD-EnKF framework through rigorous theory and systematic numerical experiments. Our main contributions are as follows:

- We develop new theoretical convergence guarantees for the large sample EnKF estimation of log-likelihood gradients in linear-Gaussian settings (Theorem 3.2).
- We combine ideas from online training of recurrent networks (specifically, truncated backpropagation through time—TBPTT) with the learning of AD-EnKF when the data sequence is long, i.e., T is large.
- We provide numerical evidence of the superior estimation accuracies of log-likelihoods and gradients afforded by EnKF relative to PF methods in high-dimensional settings. In particular, we illustrate the importance of using localization techniques, developed in the DA literature, for EnKF log-likelihood and gradient estimation, and the corresponding performance boost within AD-EnKF.
- We conduct a numerical case study of AD-EnKF on the Lorenz-96 model [60], considering parameterized dynamics, fully unknown dynamics, and correction of an inaccurate model. The importance of the Lorenz-96 model in geophysical applications and for testing the efficacy of filtering algorithms is highlighted, for instance, in [62, 53, 52, 12]. Our results show that AD-EnKF outperforms existing methods based on EM or differentiable PFs. The improvements are most significant in challenging high-dimensional and partially observed settings.

1.2. Related work. The EnKF algorithm was developed as a state estimation tool for DA [29] and is now widely used in numerical weather prediction and geophysical applications [91, 96]. Recent reviews include [42, 46, 77]. The idea behind the EnKF is to propagate N equally weighted particles through the dynamics and assimilate new observations using Kalman-type updates computed with empirical moments. When the state dimension d_x is high and the ensemble size N is moderate, traditional Kalman-type methods require $O(d_x^2)$ memory to store full covariance matrices, while storing empirical covariances in EnKFs only requires $O(Nd_x)$ memory. The use of EnKF for joint learning of state and model parameters by *state augmentation* was introduced in [3], where EnKF is run on an augmented state-space that includes the state and parameters. However, this approach requires one to design a pseudodynamic for the parameters which needs careful tuning and can be problematic when certain types of parameters (e.g., error covariance matrices) are involved [86, 23] or if the dimension of the parameters is high. In this paper, we employ EnKFs to approximate the data log-likelihood. The use of EnKF to perform derivative-free maximum likelihood estimation (MLE) is studied in [88, 71]. An empirical comparison of the likelihood computed using the EnKF and other filtering algorithms is made in [14]; see also [38, 64]. The paper [28] uses EnKF likelihood estimates to design a pseudomarginal Markov chain Monte Carlo (MCMC) method for Bayesian inference of model parameters. The works [86, 87] propose online Bayesian parameter estimation using the likelihood computed from the EnKF under a certain family of conjugate distributions. However, to the best of our knowledge, there is no prior work on state and parameter estimation that utilizes gradient information of the EnKF likelihood.

The embedding of EnKF and ensemble Kalman smoothers (EnKS) into the EM algorithm for MLE [24, 7] has been studied in [92, 93, **?**, 71], with a special focus on estimation of error covariance matrices. The expectation step (E-step) is approximated with EnKS under the Monte Carlo EM framework [95]. In addition, [12, 66] incorporate deep learning techniques in the maximization step (M-step) to train NN surrogate models. The paper [9] proposes Bayesian estimation of model error statistics, in addition to an NN emulator for the dynamics. On the other hand, [94, 17] consider online EM methods for error covariance estimation with EnKF. Although gradient information is used during the M-step to train the surrogate model [12, 66, 9], these methods do not autodifferentiate through the EnKF (see Figure 1), and accurate approximation of the E-step is hard to achieve with EnKF or EnKS.

Another popular approach for state and parameter estimation is PFs [35, 26] that approximate the filtering step by propagating samples with a kernel, reweighing them with importance sampling, and resampling to avoid weight degeneracy. PFs give an unbiased estimate of the data likelihood [21, 5]. Based upon this likelihood estimate, a particle MCMC Bayesian parameter estimation method is designed in [5]. Although PF likelihood estimates are unbiased, they suffer from two important caveats. First, their variance can be large, as they inherit the weight degeneracy of importance sampling in high dimensions [85, 10, 2, 80, 83]. Second, while the propagation and reweighing steps of PFs can be auto-differentiated, the resampling steps involve discrete distributions that cannot be handled by the reparameterization trick. For this reason, previous differentiable PFs omit autodiff of the resampling step [65, 61, 56], introducing a bias. To address this issue, the resampling step can be replaced with a differentiable optimal transport map [18], but construction of this map can be computationally expensive.

An alternative to MLE methods is to optimize a lower bound of the data log-likelihood with variational inference (VI) [7, 48, 73]. The posterior distribution over the latent states is approximated with a parametric distribution and is jointly optimized with model parameters defining the underlying state-space model (SSM). In this direction, variational sequential Monte Carlo (VSMC) methods [65, 61, 56] construct the lower bound using a PF algorithm. Moreover, the proposal distribution of the PF is parameterized and jointly optimized with model parameters defining the SSM. Although VSMC methods provide consistent data log-likelihood estimates, they suffer from the same two caveats as likelihood-based PF methods. A recent work [43] proposes blending VSMC and EnKF with an importance sampling-type lower bound estimate, which is effective if the state dimension is small. Other works that build on the VI framework include [50, 74, 31]. An important challenge is to obtain suitable parameterizations of the posterior, especially when the state dimension is high. For this reason, a restrictive Gaussian parameterization with a diagonal covariance matrix is often used in practice [50, 31].

More broadly, the development of data-driven ML frameworks for learning dynamical systems is a very active research area and we refer to [13, 36, 39, 72] for recent references that illustrate a range of techniques that do not rely on the EM algorithm, autodiff of filtering methods, or VI.

Outline. This paper is organized as follows. Section 2 formalizes our framework and reviews a characterization of the likelihood in terms of normalizing constants arising in sequential filtering. Section 3 overviews EnKF algorithms for filtering and log-likelihood estimation. Section 4 contains our main methodological contributions. Numerical experiments on linear-Gaussian and Lorenz-96 models are described in section 5. We close in section 6.

Notation. We denote by $t \in \{0, 1, \dots, T\}$ a discrete time index and by $n \in \{1, \dots, N\}$ a particle index. Time indices will be denoted with subscripts and particles with superscripts, so that x_t^n represents a generic particle at time t. We denote $x_{t_0:t_1} := \{x_t\}_{t=t_0}^{t_1}$ and $x^{n_1:n_2} := \{x^n\}_{n=n_0}^{n_1}$. The collection $x_{t_0:t_1}^{n_0:n_1}$ is defined similarly. The Gaussian density with mean m and covariance C evaluated at x is denoted by $\mathcal{N}(x; m, C)$. The corresponding Gaussian distribution is denoted by $\mathcal{N}(m, C)$. For square matrices A and B, we write $A \succ B$ if $A - B$ is positive definite, and $A \succeq B$ if $A - B$ is positive semidefinite. For $A \succeq 0$, we denote by $A^{1/2}$ the unique matrix $B \succeq 0$ such that $B^2 = A$. We denote by $|v|$ the 2-norm of a vector v and by $|A|$ the Frobenius norm of a matrix A.

2. Problem formulation.

Let $x_{0:T}$ be a time-homogeneous Markov chain of hidden *states* $x_t \in \mathbb{R}^{d_x}$ with transition kernel $p_\theta(x_t|x_{t-1})$ parameterized by $\theta \in \mathbb{R}^{d_\theta}$. Let $y_{1:T}$ be observations of the state. We seek to learn the parameter θ and recover the state process $x_{0:T}$ from the observations $y_{1:T}$. In subsection 2.1, we formalize our problem setting, emphasizing our main goal of learning unknown dynamical systems for improved DA. Subsection 2.2 describes how the log-likelihood $\mathcal{L}(\theta) = \log p_\theta(y_{1:T})$ can be written in terms of normalizing constants arising from sequential filtering. This idea will be used in section 3 to obtain EnKF estimates for $\mathcal{L}(\theta)$ and $\nabla_\theta \mathcal{L}(\theta)$, which are then employed in section 4 to learn θ by gradient ascent.

2.1. Setting and motivation. We consider the following SSM:

$$x_t = F_\alpha(x_{t-1}) + \xi_t, \qquad \xi_t \sim \mathcal{N}(0, Q_\beta), \qquad 1 \le t \le T, \tag{2.1}$$

(transition)

$$y_t = Hx_t + \eta_t, \qquad \eta_t \sim \mathcal{N}(0, R), \qquad 1 \le t \le T, \tag{2.2}$$

(observation)

$$x_0 \sim p_0(x_0). \tag{2.3}$$

(initialization)

The initial distribution p_0 and the matrices $H \in R^{d_y \times d_x}$ and $R \succ 0$ are assumed to be known. Nonlinear observations can be dealt with by augmenting the state. We further assume independence of all random variables x_0, $\xi_{1:T}$, and $\eta_{1:T}$. Finally, the transition kernel $p_\theta(x_t|x_{t-1}) = \mathcal{N}(x_t; F_\alpha(x_{t-1}), Q_\beta)$, parameterized by $\theta := \{\alpha, \beta\}$, is defined in terms of a deterministic map F_α and Gaussian additive noise. This kernel approximates an unknown state transition of the form

$$x_t = F^*(x_{t-1}) + \xi_t, \qquad \xi_t \sim \mathcal{N}(0, Q^*), \qquad 1 \le t \le T, \tag{2.4}$$

where $Q^* = 0$ if the true evolution of the state is deterministic. The parameter β allows us to estimate the possibly unknown Q^*. We consider three categories of unknown state transition F^*, leading to three types of learning problems:

(a) *Parameterized dynamics*: $F^* = F_{\alpha^*}$ is parameterized, but the true parameter α^* is unknown and needs to be estimated.

(b) *Fully unknown dynamics*: F^* is fully unknown and α represents the parameters of an NN surrogate model F_α^{NN} for F^*. The goal is to find an accurate surrogate model F_α^{NN}.

(c) *Model correction*: F^* is unknown, but an inaccurate model $F_{\mathrm{approx}} \approx F^*$ is available. Here α represents the parameters of an NN G_α^{NN} used to correct the inaccurate model. The goal is to learn α so that $F_\alpha := F_{\mathrm{approx}} + G_\alpha^{\mathrm{NN}}$ approximates F^* accurately.

In some applications, the map F^* may represent the flow between observations of an autonomous differential equation driving the state, i.e.,

$$\frac{\mathrm{d}x}{\mathrm{d}s} = f^*(x), \qquad F^* : x(s) \mapsto x(s + \Delta_s), \tag{2.5}$$

where f^* is an unknown vector field and Δ_s is the time between observations. Then, the map F_α in (2.1) (resp., F_α^{NN}, F_{approx}, G_α^{NN}) will be similarly defined as the Δ_s-flow of a differential equation with vector field f_α (resp., f_α^{NN}, f_{approx}, g_α^{NN}). Once $\theta = \{\alpha, \beta\}$ is learned, the state $x_{0:T}$ can be recovered with a filtering algorithm using the transition kernel $p_\theta(x_t|x_{t-1})$. We will illustrate the implementation and performance of AD-EnKF in these three categories of unknown dynamics in section 5 using the Lorenz-96 model to define the vector field f^*. We remark that learning NN surrogate models for the dynamics may be useful even when the true state transition F^* is known, since F_α^{NN} may be cheaper to evaluate than F^*.

Our problem setting does not require having access to a prior distribution on the parameter θ. If prior information is available, the AD-EnKF framework can seamlessly incorporate it replacing the log-likelihood with the log-posterior density in our subsequent developments. A Bayesian treatment can be appealing for unknown parameterized dynamics, where it is natural to have a priori information on the parameter. However, prior specification can be challenging for NN surrogate models.

2.2. Sequential filtering and data log-likelihood. Suppose that $\theta = \{\alpha, \beta\}$ is known. We recall that, for $1 \le t \le T$, the *filtering distributions* $p_\theta(x_t|y_{1:t})$ of the SSM (2.1)–(2.2)–(2.3) can be obtained sequentially, alternating between *forecast* and *analysis* steps:

$$\text{(forecast)} \qquad p_\theta(x_t|y_{1:t-1}) = \int \mathcal{N}(x_t; F_\alpha(x_{t-1}), Q_\beta) p_\theta(x_{t-1}|y_{1:t-1}) \mathrm{d}x_{t-1}, \tag{2.6}$$

$$\text{(analysis)} \qquad p_\theta(x_t|y_{1:t}) = \frac{1}{Z_t(\theta)} \mathcal{N}(y_t; Hx_t, R) p_\theta(x_t|y_{1:t-1}), \tag{2.7}$$

with the convention $p_\theta(\cdot|y_{1:0}) := p_\theta(\cdot)$. Here $Z_t(\theta)$ is a normalizing constant which does not depend on x_t. It can be easily shown that

$$Z_t(\theta) = p_\theta(y_t|y_{1:t-1}) = \int \mathcal{N}(y_t; Hx_t, R) p_\theta(x_t|y_{1:t-1}) \mathrm{d}x_t, \tag{2.8}$$

and therefore the data log-likelihood admits the characterization

$$\mathcal{L}(\theta) := \log p_\theta(y_{1:T}) = \sum_{t=1}^{T} \log p_\theta(y_t|y_{1:t-1}) = \sum_{t=1}^{T} \log Z_t(\theta). \tag{2.9}$$

Analytical expressions of the filtering distributions $p_\theta(x_t|y_{1:t})$ and the data log-likelihood $\mathcal{L}(\theta)$ are available only for a small class of SSMs, which includes linear-Gaussian and discrete SSMs [45, 68]. Outside these special cases, filtering algorithms need to be employed to approximate the filtering distributions, and these algorithms can be leveraged to estimate the log-likelihood.

3. Ensemble Kalman filter estimation of the log-likelihood and its gradient. In this section, we briefly review EnKFs and how they can be used to obtain an estimate $\mathcal{L}_{\mathrm{EnKF}}(\theta)$ of the log-likelihood $\mathcal{L}(\theta)$. As will be detailed in section 4, the map $\theta \mapsto \mathcal{L}_{\mathrm{EnKF}}(\theta)$ can be readily autodifferentiated to compute $\nabla_\theta \mathcal{L}_{\mathrm{EnKF}}(\theta)$, and this gradient can be used to learn the parameter θ. Subsection 3.1 gives background on EnKFs, subsection 3.2 shows how EnKFs can be used to estimate $\mathcal{L}(\theta)$, and subsection 3.3 contains novel convergence guarantees for the EnKF estimation of $\mathcal{L}(\theta)$ and $\nabla_\theta \mathcal{L}(\theta)$.

3.1. Ensemble Kalman filters. Given $\theta = \{\alpha, \beta\}$, the EnKF algorithm [29, 30] sequentially approximates the filtering distributions $p_\theta(x_t|y_{1:t})$ using N equally weighted particles $x_t^{1:N}$. At forecast steps, each particle x_t^n is propagated using the state transition equation (2.1), while at analysis steps a Kalman-type update is performed for each particle:

$$\text{(forecast step)} \qquad \widehat{x}_t^n = F_\alpha(x_{t-1}^n) + \xi_t^n, \qquad \xi_t^n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, Q_\beta), \tag{3.1}$$

$$\text{(analysis step)} \qquad x_t^n = \widehat{x}_t^n + \widehat{K}_t(y_t + \gamma_t^n - H\widehat{x}_t^n), \qquad \gamma_t^n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, R). \tag{3.2}$$

Note that the particles $x_{0:T}^{1:N}$ depend on θ, and (3.1)–(3.2) implicitly define a map $\theta \mapsto x_{0:T}^{1:N}$. The Kalman gain $\widehat{K}_t := \widehat{C}_t H^\top (H\widehat{C}_t H^\top + R)^{-1}$ is defined using the empirical covariance $\widehat{C}_t$ of the forecast ensemble $\widehat{x}_t^{1:N}$, namely

$$\widehat{C}_t = \frac{1}{N-1} \sum_{n=1}^{N} (\widehat{x}_t^n - \widehat{m}_t)(\widehat{x}_t^n - \widehat{m}_t)^\top, \quad \text{where} \quad \widehat{m}_t = \frac{1}{N} \sum_{n=1}^{N} \widehat{x}_t^n. \tag{3.3}$$

These empirical moments provide a Gaussian approximation to the *forecast distribution*

$$p_\theta(x_t|y_{1:t-1}) \approx \mathcal{N}(\widehat{m}_t, \widehat{C}_t). \tag{3.4}$$

Several implementations of EnKF are available, but for concreteness we consider only the "perturbed observation" EnKF defined in (3.1)–(3.2). In the analysis step (3.2), the observation y_t is perturbed to form $y_t + \gamma_t^n$. This perturbation ensures that in linear-Gaussian models the empirical mean and covariance of $x_t^{1:N}$ converge as $N \to \infty$ to the mean and covariance of the filtering distribution [58, 55].

3.2. Estimation of the log-likelihood and its gradient. Note from (2.9) that in order to approximate $\mathcal{L}(\theta) = \log p_\theta(y_{1:T})$, it suffices to approximate $p_\theta(y_t|y_{1:t-1})$ for $1 \le t \le T$. Now, using (2.8) and the EnKF approximation (3.4) to the forecast distribution, we obtain

$$p_\theta(y_t|y_{1:t-1}) \approx \int \mathcal{N}(y_t; Hx_t, R)\mathcal{N}(x_t; \widehat{m}_t, \widehat{C}_t)\mathrm{d}x_t = \mathcal{N}\big(y_t; H\widehat{m}_t, H\widehat{C}_tH^\top + R\big). \tag{3.5}$$

Therefore, we have the following estimate of the data log-likelihood:

$$\mathcal{L}_{\mathrm{EnKF}}(\theta) := \sum_{t=1}^{T} \log \mathcal{N}\big(y_t; H\widehat{m}_t, H\widehat{C}_tH^\top + R\big) \approx \mathcal{L}(\theta). \tag{3.6}$$

Notice that the forecast empirical moments $\{\widehat{m}_t, \widehat{C}_t\}_{t=1}^T$, and hence $\mathcal{L}_{\mathrm{EnKF}}(\theta)$, depend on θ in two distinct ways. First, each forecast particle $\widehat{x}_t^n$ in (3.1) depends on a particle x_t^n, which indirectly depends on θ. Second, each forecast particle depends on $\theta = \{\alpha, \beta\}$ directly through F_α and Q_β. The estimate $\mathcal{L}_{\mathrm{EnKF}}(\theta)$ can be computed online with EnKF and is stochastic as it depends on the randomness used to propagate the particles, e.g., the choice of random seed. The whole procedure is summarized in Algorithm 3.1, which implicitly defines a stochastic map $\theta \mapsto \mathcal{L}_{\mathrm{EnKF}}(\theta)$. Before discussing the autodiff of this map and learning of the parameter θ in section 4, we establish the large ensemble convergence of $\mathcal{L}_{\mathrm{EnKF}}(\theta)$ and $\nabla_\theta \mathcal{L}_{\mathrm{EnKF}}(\theta)$ toward $\mathcal{L}(\theta)$ and $\nabla_\theta \mathcal{L}(\theta)$ in a linear setting.

Algorithm 3.1. Ensemble Kalman filter and log-likelihood estimation

Input: $\theta = \{\alpha, \beta\}, y_{1:T}, x_0^{1:N}$. (If $x_0^{1:N}$ is not specified, draw $x_0^n \overset{\text{i.i.d.}}{\sim} p_0(x_0)$.)
Initialize $\mathcal{L}_{\mathrm{EnKF}}(\theta) = 0$.
for $t = 1, \dots, T$ **do**
  Set $\widehat{x}_t^n = F_\alpha(x_{t-1}^n) + \xi_t^n$, where $\xi_t^n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, Q_\beta)$. ▷ Forecast step
  Compute $\widehat{m}_t, \widehat{C}_t$ by (3.3) and set $\widehat{K}_t = \widehat{C}_tH^\top(H\widehat{C}_tH^\top + R)^{-1}$.
  Set $x_t^n = \widehat{x}_t^n + \widehat{K}_t(y_t + \gamma_t^n - H\widehat{x}_t^n)$, where $\gamma_t^n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, R)$. ▷ Analysis step
  Set $\mathcal{L}_{\mathrm{EnKF}}(\theta) \leftarrow \mathcal{L}_{\mathrm{EnKF}}(\theta) + \log \mathcal{N}\big(y_t; H\widehat{m}_t, H\widehat{C}_tH^\top + R\big)$.
end for
Output: EnKF particles $x_{0:T}^{1:N}$. Log-likelihood estimate $\mathcal{L}_{\mathrm{EnKF}}(\theta)$.

3.3. Large sample convergence: Linear setting. In this section we consider a linear setting and provide large N convergence results for the log-likelihood estimate $\mathcal{L}_{\mathrm{EnKF}}(\theta)$ and its gradient $\nabla_\theta \mathcal{L}_{\mathrm{EnKF}}(\theta)$ toward $\mathcal{L}(\theta)$ and $\nabla_\theta \mathcal{L}(\theta)$ for any given θ, for a fixed data sequence $y_{1:T}$. The mappings $\mathcal{L}$ and $\mathcal{L}_{\mathrm{EnKF}}$ are defined in (2.9) and (3.6), respectively. For notation convenience, we drop θ in the function argument, since the main dependence will be on N in this section. Similar to [58, 51], we study L^p convergence for any $p \geq 1$.

Theorem 3.1. *Assume that the state transition (2.1) is linear, i.e.,*

$$x_t = A_\alpha x_{t-1} + \xi_t, \qquad \xi_t \sim \mathcal{N}(0, Q_\beta), \qquad A_\alpha \in \mathbb{R}^{d_x \times d_x}, \tag{3.7}$$

and that the initial distribution p_0 is Gaussian. Then, for any $\theta = \{\alpha, \beta\}$ and for any $p \geq 1$, $\mathcal{L}_{EnKF}$ converges to $\mathcal{L}$ in L^p with rate $1/\sqrt{N}$, i.e.,

$$\left(\mathbb{E}\left|\mathcal{L}_{EnKF} - \mathcal{L}\right|^p\right)^{1/p} \leq cN^{-1/2}, \tag{3.8}$$

where c does not depend on N but may depend on θ, d_x, and d_y.

The linearity of the flow $F_\alpha(\cdot)$ is equivalent to the linearity of the vector field $f_\alpha(\cdot)$. Although the convergence of the EnKF to the KF in linear settings has been studied in DA [58, 55, 51, 22] and in filtering approaches to inverse problems [84, 15], there are no existing convergence results for EnKF log-likelihood estimation. Two related works are [47], which provides a heuristic argument for convergence in the case $T = 1$, and [19], where a continuous-time version of EnKF is considered.

Most of the theoretical analysis of EnKF is based on the *propagation of chaos* statement [63, 90]: EnKF defines an interacting particle system, where the interaction is through the empirical mean $\widehat{m}_t$ and covariance matrix $\widehat{C}_t$ of the forecast ensemble $\widehat{x}_t^{1:N}$. As $N \to \infty$, one hopes that these empirical moments can be replaced by their deterministic limits, and that the particles will hence evolve independently. The large N limits of $\widehat{m}_t, \widehat{C}_t$ turn out to be the mean and covariance matrix of the KF forecast distribution. We will leave the construction of the propagation of chaos statement as well as the proof of Theorem 3.1 to Appendix A.

Since this paper focuses on gradient based approaches to the learning of $\theta = \{\alpha, \beta\}$, it is thus interesting to compare the gradient $\nabla_\theta \mathcal{L}_{\mathrm{EnKF}}$ to the true gradient $\nabla_\theta \mathcal{L}$, as $N \to \infty$, if both of them exist. The intuition is that if $\nabla_\theta \mathcal{L}_{\mathrm{EnKF}}$ is an accurate estimate of $\nabla_\theta \mathcal{L}$, then one can perform gradient-based optimization over $\mathcal{L}_{\mathrm{EnKF}}$ as if one were directly optimizing over the true log-likelihood $\mathcal{L}$. For the gradient w.r.t. β to be well defined, we write $S_\beta = Q_\beta^{1/2}$ in the following statement, so that β does not appear in the stochasticity of the algorithm. This is also known as the "reparameterization trick," which will be discussed later in subsection 4.1.

Theorem 3.2. *Assume that the state transition (2.1) is linear, i.e.,*

$$x_t = A_\alpha x_{t-1} + S_\beta \xi_t, \qquad \xi_t \sim \mathcal{N}(0, I_{d_x}), \qquad A_\alpha \in \mathbb{R}^{d_x \times d_x}, \tag{3.9}$$

and that the initial distribution p_0 is Gaussian. Assume the parameterizations $\alpha \mapsto A_\alpha$ and $\beta \mapsto S_\beta$ are differentiable. Then, for any $\theta = \{\alpha, \beta\}$, both $\nabla_\theta \mathcal{L}_{EnKF}$ and $\nabla_\theta \mathcal{L}$ exist and, for any $p \geq 1$, $\nabla_\theta \mathcal{L}_{EnKF}$ converges to $\nabla_\theta \mathcal{L}$ in L^p with rate $1/\sqrt{N}$, i.e.,

$$\left(\mathbb{E}\left|\nabla_\theta \mathcal{L}_{EnKF} - \nabla_\theta \mathcal{L}\right|^p\right)^{1/p} \leq cN^{-1/2}, \tag{3.10}$$

where c does not depend on N but may depend on θ, d_x, and d_y.

An important observation is that θ only enters the objective function $\mathcal{L}_{\mathrm{EnKF}}$ through the empirical mean $\widehat{m}_t$ and covariance matrix $\widehat{C}_t$ of the forecast ensemble. As $N \to \infty$, one hopes that these empirical moments can be replaced by their deterministic limits, and gradients based on these empirical moments can be replaced by gradients based on their deterministic limits. The gradients taken in the limits turn out to be those of the true log-likelihood $\mathcal{L}$. Again, the proof relies on the propagation of chaos statement and is left to Appendix B.

Theorem 3.2 should be compared with log-likelihood gradient estimation with PFs. The paper [18] shows that the gradient $\nabla_\theta \mathcal{L}_{\mathrm{PF}}$ of PF log-likelihood estimate is biased, even in the linear setting, if one ignores the gradient from resampling steps, which is the method used in practice [65, 61, 56].

4. Autodifferentiable ensemble Kalman filters.

This section contains our main methodological contributions. We introduce our AD-EnKF framework in subsection 4.1. We then describe in subsection 4.2 how to handle long observation data, i.e., large T, using TBPTT. In subsection 4.3, we highlight how various techniques introduced for EnKF in the DA community, e.g., localization and covariance inflation, can be incorporated into our framework. Finally, subsection 4.4 discusses the computational and memory costs.

4.1. Main algorithm.

Our core method is shown in Algorithm 4.1, and our PyTorch implementation is at https://github.com/ymchen0/torchEnKF. The gradient of the stochastic map $\theta^k \mapsto \mathcal{L}_{\mathrm{EnKF}}(\theta^k)$ can be evaluated using autodiff libraries [70, 11, 1]. More specifically, reverse-mode autodiff can be performed for common matrix operations like matrix multiplication, inverse, and determinant [34]. We use the "reparameterization trick" [48, 76] to autodifferentiate through the stochasticity in the EnKF algorithm. Specifically, in Algorithm 3.1, line 3, we draw ξ_t^n from a distribution $\mathcal{N}(0, Q_\beta)$ that involves a parameter β with respect to which we would like to compute the gradient. For this operation to be compatible with the autodiff, we reparameterize

(4.1)
$$\widehat{x}_t^n = F_\alpha(x_t^n) + \xi_t^n \quad \xi_t^n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, Q_\beta) \quad \Longleftrightarrow \quad \widehat{x}_t^n = F_\alpha(x_t^n) + Q_\beta^{1/2}\xi_t^n \quad \xi_t^n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_{d_x}),$$

so that the gradient with respect to β admits an unbiased estimate. In contrast to the EnKF, the resampling step of PFs cannot be readily autodifferentiated [65, 61, 56]. The algorithm can be stopped when certain convergence criteria have been met, e.g., when the relative change

Algorithm 4.1. Autodifferentiable ensemble Kalman filter (AD-EnKF)

Input: Observations $y_{1:T}$. Learning rate η.
1: **Initialize** SSM parameter θ^0 and set $k = 0$.
2: **while** not converging **do**
3: $x_{0:T}^{1:N}, \mathcal{L}_{\mathrm{EnKF}}(\theta^k) = \textsc{EnsembleKalmanFilter}(\theta^k, y_{1:T})$. ▷ Alg. 3.1
4: Compute $\nabla_\theta \mathcal{L}_{\mathrm{EnKF}}(\theta^k)$ by autodifferentiating the map $\theta^k \mapsto \mathcal{L}_{\mathrm{EnKF}}(\theta^k)$.
5: Set $\theta^{k+1} = \theta^k + \eta \nabla_\theta \mathcal{L}_{\mathrm{EnKF}}(\theta^k)$ and $k \leftarrow k + 1$.
6: **end while**
Output: Learned SSM parameter θ^k and EnKF particles $x_{0:T}^{1:N}$.

in the 10-step moving average of the EnKF log-likelihood $\mathcal{L}_{\mathrm{EnKF}}(\theta^k)$ does not exceed a prespecified threshold of 10^{-2}. In our numerical experiments in section 5, we run the algorithm for at least 50 additional iterations past convergence to demonstrate its long-time performance and stability.

4.2. Truncated gradients for long sequences. If the sequence length T is large, although $\mathcal{L}_{\mathrm{EnKF}}(\theta)$ and its gradient $\nabla_\theta \mathcal{L}_{\mathrm{EnKF}}(\theta)$ can be evaluated using the aforementioned techniques, the practical value of Algorithm 4.1 is limited for two reasons. First, computing these quantities requires a full filtering pass of the data, which may be computationally costly. Moreover, for the gradient ascent methods to achieve a good convergence rate, multiple evaluations of gradients are often needed, requiring an equally large number of filtering passes. The second reason is that, like recurrent networks, Algorithm 4.1 may suffer from exploding or vanishing gradients [69] as the derivatives are multiplied together using chain rules in the backpropagation.

Our proposed technique can address both of these issues by borrowing the ideas of TBPTT from the recurrent NN literature [97, 89] and the recursive maximum likelihood method from the hidden Markov models literature [57]. The idea is to divide the sequence into subsequences of length L. Instead of computing the log-likelihood of the whole sequence and then backpropagating, one computes the log-likelihood of each subsequence and backpropagates within that subsequence. The subsequences are processed sequentially, and the EnKF output of the previous subsequence (i.e., the location of particles) are used as the input to the next subsequence. In this way, one performs $\lceil T/L \rceil$ gradient updates in a *single* filtering pass, and since the gradients are backpropagated across a time span of length at most L, gradient explosion/vanishing is more unlikely to happen. This approach is detailed in Algorithm 4.2.

4.3. Localization for high state dimensions. In practice, the state often represents a physical quantity that is discretized in spatial coordinates (e.g., numerical solution to a time-evolving PDE), which leads to a high state dimension d_x. In order to reduce the computational and memory complexity, EnKF is often run with $N < d_x$. A small ensemble size N causes rank deficiency of the forecast sample covariance $\widehat{C}_t$, which may cause spurious correlations between spatial coordinates that are far apart. In other words, for (i, j) such that $|i - j|$

Algorithm 4.2. AD-EnKF with truncated backprop (AD-EnKF-T)

Input: Observations $y_{1:T}$. Learning rate η. Subsequence length L.
Initialize SSM parameter θ^0 and set $k = 0$.
while not converging **do**
  Set $x_0^n \overset{\text{i.i.d.}}{\sim} p_0(x_0)$.
  for $j = 0, \dots, T/L - 1$ **do**
    Set $t_0 = jL$, $t_1 = \min\{(j+1)L, T\}$.
    $x_{t_0:t_1}^{1:N}, \mathcal{L}_{\mathrm{EnKF}}(\theta^k) = \textsc{EnsembleKalmanFilter}(\theta^k, y_{(t_0+1):t_1}, x_{t_0}^{1:N})$. ▷ Alg. 3.1
    Set $\theta^{k+1} = \theta^k + \eta \nabla_\theta \mathcal{L}_{\mathrm{EnKF}}(\theta^k)$ and $k \leftarrow k + 1$.
  end for
end while
Output: Learned SSM parameter θ^k and EnKF particles $x_{0:T}^{1:N}$.

is large, the (i,j)th coordinate of $\widehat{C}_t$ may not be close to 0, although one would expect it to be small since it represents the correlation between spatial locations that are far apart. This problem can be addressed using localization techniques, and we shall focus on *covariance tapering* [40]. The idea is to "taper" the forecast sample covariance matrix $\widehat{C}_t$ so that the nonzero spurious correlations are zeroed out. This method is implemented defining a $d_x \times d_x$ matrix ρ with 1's on the diagonal and entries smoothly decaying to 0 off the diagonal, and replacing the forecast sample covariance matrix $\widehat{C}_t$ in Algorithm 3.1 by $\rho \circ \widehat{C}_t$, where $\circ$ denotes the elementwise matrix product. Common choices of ρ were introduced in [33]. Covariance tapering can be easily adopted within our AD-EnKF framework. We find that covariance tapering not only stabilizes the filtering procedure, which had been noted before, e.g., [41, 37], but also helps to obtain low-variance estimates of the log-likelihood and its gradient—see the discussion in subsubsection 5.1.2. Localization techniques relying on local serial updating of the state [41, 67, 79] could also be considered.

Another useful tool for EnKF with $N < d_x$ is *covariance inflation* [4], which prevents the ensemble from collapsing toward its mean after the analysis update [32]. In practice, this can be performed by replacing the forecast sample covariance matrix $\widehat{C}_t$ in Algorithm 3.1 by $(1+\zeta)\widehat{C}_t$, where $\zeta > 0$ is a small constant that needs to be tuned. Although not considered in our experiments, covariance inflation can also be easily adopted within our AD-EnKF framework.

4.4. Computation and memory costs. Autodifferentiation of the map $\theta^k \mapsto \mathcal{L}_{\mathrm{EnKF}}(\theta^k)$ in Algorithm 4.1 does not introduce an extra order of computational cost compared to the evaluation of this map alone. Thus, the computational cost of AD-EnKF is at the same order as that of a standard EnKF. The computation cost of EnKF can be found in, e.g., [77]. Moreover, AD-EnKF can be parallelized and speeded up with a GPU.

Like a standard EnKF, when no covariance tapering is applied, AD-EnKF has $O(Nd_x)$ memory cost since it does not explicitly compute the sample covariance matrix $\widehat{C}_t$[1]. With covariance tapering, the memory cost is at most $O(\max\{N,r\}d_x)$, where r is the tapering radius, if the tapering matrix ρ is sparse with $O(rd_x)$ nonzero entries. This sparsity condition is satisfied when using common tapering matrices [33]. In terms of the time dimension, the memory cost of AD-EnKF can be reduced from $O(T)$ to $O(L)$ with the TBPTT in subsection 4.2. Unlike previous work on EM-based approaches [12, 9, 71], where the locations of all particles $x_t^{1:N}$ across the whole time span of T need to be stored, AD-EnKF-T only requires storing the particles within a time span of L to perform a gradient step.

If the transition map F_α is defined by the flow map of an ODE with vector field f_α, we can use adjoint methods to differentiate efficiently through F_α in the forecast step (3.1). Use of the adjoint method is facilitated by NeuralODE autodiff libraries [16] that have become an important tool for learning continuous-time dynamical systems [6, 78, 20]. Instead of discretizing F_α with a numerical solver applied to f_α and differentiating through the solver's steps as in [12, 9], we directly differentiate through F_α by solving an adjoint differential equation, which does not require us to store all intermediate steps from the numerical solver, reducing the memory cost. More details can be found in [16], and the PyTorch package

[1]Note that $\widehat{C}_t H^\top = \frac{1}{N-1}\sum_{n=1}^N (\widehat{x}_t^n - \widehat{m}_t)(H\widehat{x}_t^n - H\widehat{m}_t)^\top$ and $H\widehat{C}_t H^\top = \frac{1}{N-1}\sum_{n=1}^N (H\widehat{x}_t^n - H\widehat{m}_t)(H\widehat{x}_t^n - H\widehat{m}_t)^\top$, which require $O(d_x \max\{N, d_y\})$ and $O(d_y \max\{N, d_y\})$ memory, respectively. Both of them are less than $O(d_x^2)$ if $d_y \ll d_x$ and $N \ll d_x$.

provided by the authors can be incorporated within our AD-EnKF framework with minimal effort.

5. Numerical experiments.

5.1. Linear-Gaussian model. In this section, we focus on parameter estimation in a linear-Gaussian model with a banded structure on model dynamic and model error covariance matrix. This experiment falls into the category of "parameterized dynamics" in subsection 2.1. We first illustrate the convergence results of the log-likelihood estimate $\mathcal{L}_{\text{EnKF}}$ and gradient estimate $\nabla_\theta \mathcal{L}_{\text{EnKF}}$ presented in subsection 3.3, since the true values $\mathcal{L}$ and $\nabla_\theta \mathcal{L}$ are available in closed form. We also show that the localization techniques described in subsection 4.3 lead to a more accurate estimate when the ensemble size is small. Finally, we show that having a more accurate estimate, especially for the gradient, improves the parameter estimation.

We compare the EnKF to PF methods. Similar to the EnKF, the PF also provides an estimate of the log-likelihood and its gradient. Different from [65, 61, 56], we adopt the PF with optimal proposal [26] as it is implementable for the family of SSMs considered in this paper [25, 82], and we find it to be more stable than separately training a variational proposal. To compute the log-likelihood gradient for the PF, we follow the same strategy as in [65, 61, 56] and do not differentiate through the resampling step. The full algorithm, which we abbreviate as AD-PF, is presented in section SM2.

We consider the following SSM, similar to [98, 87]:

$$x_t = A_\alpha x_{t-1} + \xi_t, \qquad \xi_t \sim \mathcal{N}(0, Q_\beta), \qquad 1 \le t \le T, \tag{5.1a}$$

$$y_t = H x_t + \eta_t, \qquad \eta_t \sim \mathcal{N}(0, 0.5 I_{d_y}), \qquad 1 \le t \le T, \tag{5.1b}$$

$$x_0 \sim \mathcal{N}(0, 4 I_{d_x}), \tag{5.1c}$$

where

$$A_\alpha = \begin{bmatrix} \alpha_1 & \alpha_2 & & 0 \\ \alpha_3 & \alpha_1 & \ddots & \\ & \ddots & \ddots & \alpha_2 \\ 0 & & \alpha_3 & \alpha_1 \end{bmatrix}, \qquad [Q_\beta]_{i,j} = \beta_1 \exp(-\beta_2 |i-j|). \tag{5.1d}$$

Here $[Q_\beta]_{i,j}$ denotes the (i,j)th entry of Q_β. Intuitively, β_1 controls the scale of error, while β_2 controls how error is correlated across spatial coordinates. We set $\alpha = (\alpha_1, \alpha_2, \alpha_3)$, $\beta = (\beta_1, \beta_2)$, and $\theta = \{\alpha, \beta\}$.

5.1.1. Estimation accuracy of $\mathcal{L}_{\text{EnKF}}$ and $\nabla_\theta \mathcal{L}_{\text{EnKF}}$. As detailed above, a key idea proposed in this paper is to estimate $\mathcal{L}(\theta)$, $\nabla_\alpha \mathcal{L}(\theta)$, and $\nabla_\beta \mathcal{L}(\theta)$ with quantities $\mathcal{L}_{\text{EnKF}}(\theta)$, $\nabla_\alpha \mathcal{L}_{\text{EnKF}}(\theta)$ and $\nabla_\beta \mathcal{L}_{\text{EnKF}}(\theta)$ obtained by running an EnKF and differentiating through its computations using autodiff. Since these estimates will be used by AD-EnKF to perform gradient ascent, it is critical to assess their accuracy. We do so in this section for a range of values of θ.

We first simulate observation data $y_{1:T}$ from the true model with $d_x = d_y \in \{20, 40, 80\}$, $T = 10$, $H = I_{d_x}$, $\alpha^* = (0.3, 0.6, 0.1)$, and $\beta^* = (0.5, 1)$. Given data $y_{1:T}$, the true data log-likelihood $\mathcal{L}(\theta) = p_\theta(y_{1:T})$ and gradient $\nabla_\theta \mathcal{L}(\theta)$, which can be decomposed into $\nabla_\alpha \mathcal{L}(\theta)$,

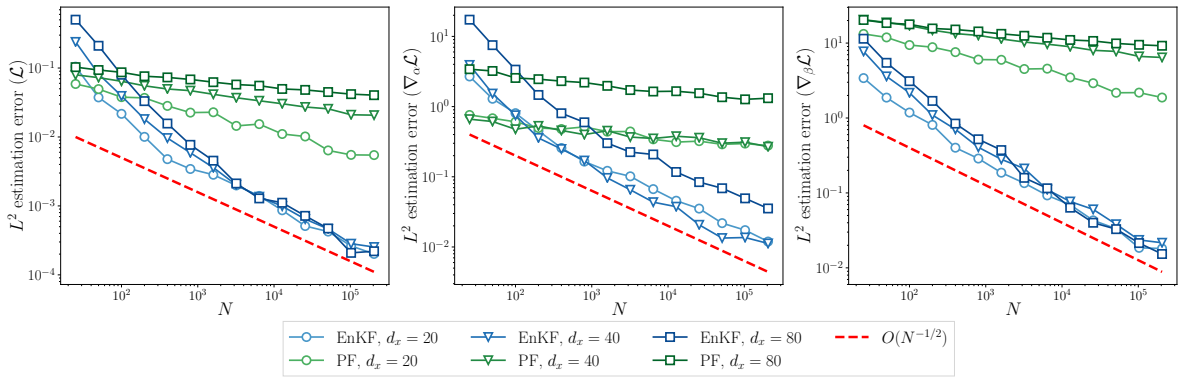


Figure 2. *Relative L^2 estimation errors of the log-likelihood (left) and its gradient w.r.t. α (middle) and β (right), computed using EnKF and PF, as a function of N, for the linear-Gaussian model (5.1). State dimension $d_x \in \{20, 40, 80\}$. θ is evaluated at the true parameters $\{\alpha^*, \beta^*\}$ (subsubsection 5.1.1).*

$\nabla_\beta \mathcal{L}(\theta)$, can be computed analytically. We perform $P = 50$ EnKF runs, and report a Monte Carlo estimate of the relative L^2 errors of the log-likelihood and gradient estimates (see section SM4 for their definition) as the ensemble size N increases. Figure 2 shows the results when θ is evaluated at the true parameters $\{\alpha^*, \beta^*\}$. Intuitively, this θ is close to optimal since it is the one that generates the data. We also show in Figure SM1 in section SM1 the results when θ is evaluated at a parameter that is not close to optimal: $\alpha = (0.5, 0.5, 0.5), \beta = (1, 0.1)$. Both figures illustrate that the relative L^2 estimation errors of the log-likelihood and its gradient computed using EnKF converge to zero at a rate of approximately $N^{-1/2}$. Moreover, the state dimension d_x has a small empirical effect on the convergence rate. On the other hand, those computed using PF have a slower convergence rate or barely converge, especially for the gradient (see the third plot in Figure SM1). We recall that the resampling parts are discarded from the autodiff of PFs, which introduces a bias. Moreover, the empirical convergence rate is slightly slower in higher state dimensions. Comparing the estimation error of EnKF and PF under the same d_x choice, we find that when the number of particles is large (>500), EnKF gives a more accurate estimate than PF. However, when the number of particles is small, EnKF is less accurate, but we will show in the next section how the EnKF results can be significantly improved using localization techniques. Unreported experimental results suggest that the relative L^2 error in the EnKF estimation of the log-likelihood and its gradient increase linearly with T for a fixed ensemble size N.

5.1.2. Effect of localization. In practice, for computational and memory concerns, the number of particles used for EnKF is typically small (<100), and hence it is necessary to get an accurate estimate of log-likelihood and its gradients using a small number of particles. We use the covariance tapering techniques discussed in subsection 4.3, where $\widehat{C}_t$ is replaced by $\rho \circ \widehat{C}_t$ in Algorithm 3.1, and ρ is defined using the fifth-order piecewise polynomial correlation function of Gasperi and Cohn [33]. The detailed construction of ρ is left to section SM4, with a hyperparameter r that controls the tapering radius.

Figure 3 shows the estimation results when the state dimension is set to be $d_x = 80$ and θ is evaluated at (α^*, β^*), while different tapering radii r are applied. The plots of EnKF with

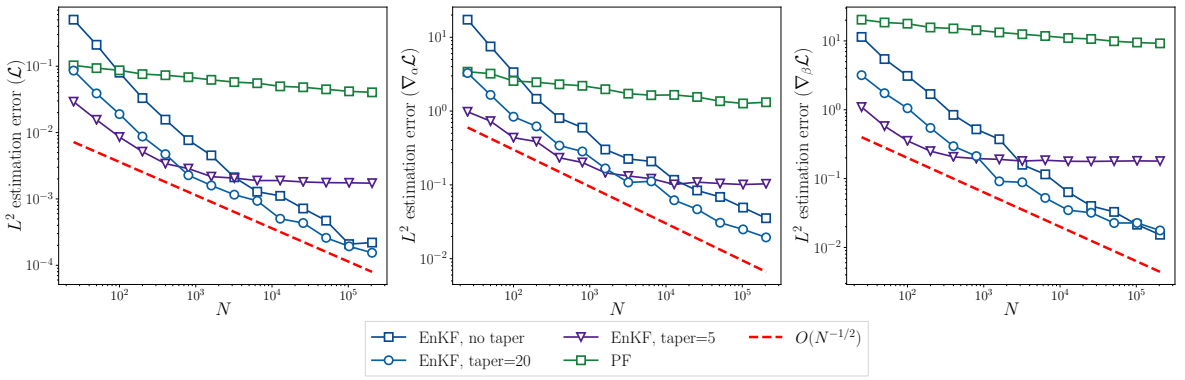


Figure 3. *Relative L^2 estimation errors of log-likelihood (left) and its gradient w.r.t. α (middle) and β (right), computed using EnKF and PF, with different covariance tapering radius applied to EnKF for the linear-Gaussian model (5.1). State dimension $d_x = 80$. θ is evaluated at the true parameters $\{\alpha^*, \beta^*\}$ (subsubsection 5.1.2).*

no tapering and the plots of PF are the same as in Figure 2. We find that covariance tapering can reduce the estimation error of the log-likelihood and its gradient when the number of particles is small. Moreover, having a smaller tapering radius leads to a better estimation when the number of particles is small. As the number of particles grows larger, covariance tapering may worsen the estimation of both log-likelihood and its gradient. This is because the sampling error and spurious correlation that occurs in the sample covariance matrix in EnKF will be overcome by a large number of particles, and hence covariance tapering will only act as a modification to the objective function $\mathcal{L}_{\text{EnKF}}$, leading to inconsistent estimates. However, there is no reason for using localization when one can afford a large number of particles. When computational constraints require fewer particles than state dimension, we find that covariance tapering not only is beneficial to the parameter estimation problems but is also beneficial to learning of the dynamics in high dimensions, as we will show in later sections. Results when θ is evaluated at parameters that are not optimal ($\alpha = (0.5, 0.5, 0.5)$, $\beta = (1, 0.1)$) are shown in Figure SM2 in section SM1, where the beneficial effect of tapering is evident.

5.1.3. Parameter learning. Here we illustrate how the estimation accuracy of the log-likelihood and its gradient, especially the latter, affect the parameter learning with AD-EnKF. Since our framework relies on gradient-based learning of parameters, intuitively, the less biased the gradient estimate is, the closer our learned parameter will be to the true MLE solution.

We first consider the setting where the state dimension is set to be $d_x = 80$. We run AD-EnKF for 1000 iterations with gradient ascent under the following choices of ensemble size and tapering radius: (1) $N = 1000$ with no tapering; (2) $N = 50$ with no tapering; and (3) $N = 50$ with tapering radius 5. We also run AD-PF with $N = 1000$ particles. Throughout, one "training iteration" corresponds to processing the whole data sequence once. Additional implementation details are available in the appendices. Figures 4 and 5 show a single run of parameter learning under each setting, where we include for reference the MLE obtained by running gradient ascent until convergence with the true gradient $\nabla_\theta \mathcal{L}$ (denoted with the red dashed line). The objective function, i.e., the likelihood estimates $\mathcal{L}_{\text{EnKF}}$ and $\mathcal{L}_{\text{PF}}$, are

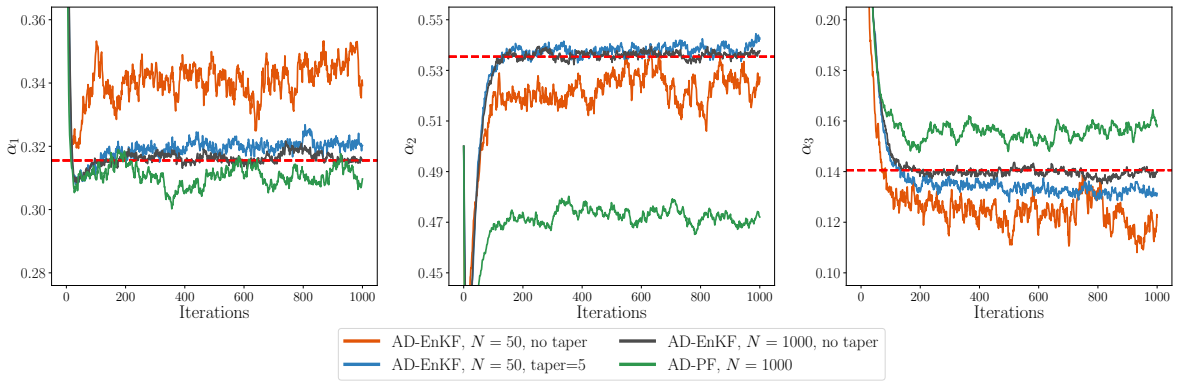


Figure 4. *Learned parameter α as a function of training iterations for the linear-Gaussian model (5.1). State dimension $d_x = 80$. Red dashed lines are the MLE solutions to the true data log-likelihood $\mathcal{L}$. Our proposed AD-EnKF method with covariance tapering achieves a lower estimation error with $N = 50$ particles than AD-PF with $N = 1000$ (subsubsection 5.1.3).*

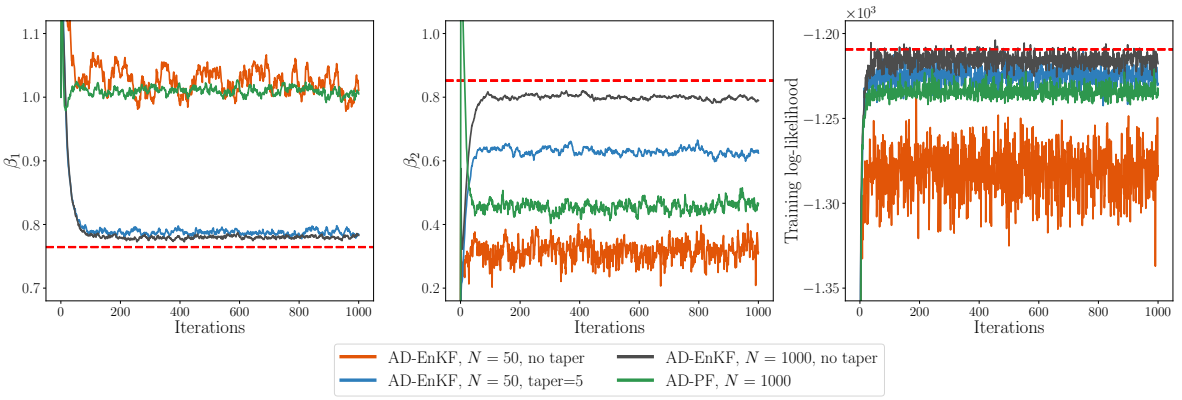


Figure 5. *Learned parameter β, and training objective $\mathcal{L}_{EnKF}$, $\mathcal{L}_{PF}$ as a function of training iterations for the linear-Gaussian model (5.1). Red dashed lines are the MLE solutions to the true data log-likelihood $\mathcal{L}$ (left and middle), and the maximum value attained by $\mathcal{L}$ (right). Our proposed AD-EnKF method with covariance tapering achieves a lower estimation error with $N = 50$ particles than AD-PF with $N = 1000$ (subsubsection 5.1.3).*

also plotted as a function of training iterations. Results with other choices of state dimension d_x are summarized in Table 5.1, where we take the values of α at the final iteration and compute their distance to the true MLE solution. The procedure is repeated 10 times, and the mean and standard deviations are reported. The results all show a similar trend: AD-EnKF with $N = 1000$ particles performs the best (small errors and small fluctuations) for all settings, while AD-EnKF with $N = 50$ particles and covariance tapering performs second best. AD-EnKF with $N = 50$ without covariance tapering comes in at third place, and the AD-PF method performs the worst, indicating the superiority of the AD-EnKF method to the AD-PF method for high-dimensional linear-Gaussian models of the form (5.1) and the utility of localization techniques. Importantly, the findings here are consistent with the plots in Figure 3. This behavior is in agreement with the intuition that the estimation accuracy of the log-likelihood gradient determines the parameter learning performance.

Table 5.1

Euclidean distance ($\times 10^{-2}$) from the learned parameter α at the final iteration to the true MLE solution, under varying dimensional settings for the linear-Gaussian model (5.1). The parameter values recovered by our proposed AD-EnKF method with covariance tapering and $N = 50$ are closer to the MLE solution than the ones recovered by AD-PF with $N = 1000$ (subsubsection 5.1.3).

	$d_x = 20$ $N = 50$	$d_x = 20$ $N = 1000$	$d_x = 40$ $N = 50$	$d_x = 40$ $N = 1000$	$d_x = 80$ $N = 50$	$d_x = 80$ $N = 1000$
AD-EnKF (no taper)	1.65 ±0.30	0.07 ±0.06	4.12±0.73	0.17±0.09	4.14±0.67	0.20±0.14
AD-EnKF (taper=5)	0.53±0.18	–	0.35±0.27	–	1.05±0.38	–
AD-PF	7.75±0.37	3.51±0.35	8.58±0.25	5.59±0.31	9.28±0.49	6.77±0.24

5.2. Lorenz-96. In this section, we illustrate our AD-EnKF framework in the three types of learning problems mentioned in subsection 2.1: parameterized dynamics, fully unknown dynamics, and model correction. We will compare our method to AD-PF, as in subsection 5.1. We will also compare our method to the EM-EnKF method implemented in [9, 12], which we abbreviate as EM and which is detailed in section SM3. We emphasize that the gradients computed in the EM are different from the ones computed in AD-EnKF and in particular do not auto-differentiate through the EnKF.

The reference Lorenz-96 model [60] is defined by (2.5) with vector field

$$f^{*(i)}(x) = -x^{(i-1)}(x^{(i-2)} - x^{(i+1)}) - x^{(i)} + 8, \qquad 0 \le i \le d_x - 1, \tag{5.2}$$

where $x^{(i)}$ and $f^{*(i)}$ are the ith coordinate of x and component of f^*. By convention $x^{(-1)} := x^{(d_x-1)}, x^{(-2)} := x^{(d_x-2)}$, and $x^{(d_x)} := x^{(0)}$. We assume there is no noise in the reference state transition model, i.e., $Q^* = 0$. The goal is to recover the reference state transition model with $p_\theta(x_t|x_{t-1}) = \mathcal{N}(x_t; F_\alpha(x_{t-1}), Q_\beta)$ from the data $y_{1:T}$, where F_α is the flow map of a vector field f_α, and then recover the states $x_{1:T}$. The parameterized error covariance Q_β in the transition model is assumed to be diagonal, i.e., $Q_\beta = \mathrm{diag}(\beta)$ with $\beta \in \mathbb{R}^{d_x}$. The parameterized vector field f_α is defined differently for the three types of learning problems, as we lay out below. We quantify performance using the forecast error (RMSE-f), the analysis/filter error (RMSE-a), and the test log-likelihood. These metrics are defined in section SM4.

5.2.1. Parameterized dynamics. We consider the same setting as in [8], where

$$\begin{aligned} f_\alpha^{(i)}(x) = \big[&1, x^{(i-2)}, x^{(i-1)}, x^{(i)}, x^{(i+1)}, x^{(i+2)}, \\ &\big(x^{(i-2)}\big)^2, \big(x^{(i-1)}\big)^2, \big(x^{(i)}\big)^2, \big(x^{(i+1)}\big)^2, \big(x^{(i+2)}\big)^2, \\ &x^{(i-2)}x^{(i-1)}, x^{(i-1)}x^{(i)}, x^{(i)}x^{(i+1)}, x^{(i+1)}x^{(i+2)}, \\ &x^{(i-2)}x^{(i)}, x^{(i-1)}x^{(i+1)}, x^{(i)}x^{(i+2)}\big]^\top \alpha, \qquad 0 \le i \le d_x - 1, \end{aligned} \tag{5.3}$$

and $\alpha \in \mathbb{R}^{18}$ is interpreted as the coefficients of some "basis polynomials" representing the governing equation of the underlying system. The parameterized governing equation of the ith coordinate depends on its $N_1 = 5$ neighboring coordinates, and the second-order polynomials involve only interactions between coordinates that are at most $N_2 = 2$ indices apart. The reference ODE (5.2) satisfies $f^* = f_{\alpha^*}$, where $\alpha^* \in \mathbb{R}^{18}$ has nonzero entries

$$\alpha_0^* = 8, \quad \alpha_3^* = -1, \quad \alpha_{11}^* = -1, \quad \alpha_{16}^* = 1, \tag{5.4}$$

and zero entries otherwise. Here the dimension of $\theta = \{\alpha, \beta\}$ is $d_\theta = 18 + d_x$.

We first consider the specific case with $d_x = d_y = 40$, $H = I_{40}$. We set $R = I_{40}$ and $x_0 \sim \mathcal{N}(0, 50I_{40})$. We generate four sequences of training data with the reference model for $T = 300$ with time between consecutive observations $\Delta_s = 0.05$. Both flow maps F^* and F_α are integrated using a fourth-order Runge–Kutta (RK4) method with step size $\Delta_s^{\text{int}} = 0.01$, with adjoint methods implemented for backpropagation through the ODE solver [16].

We use AD-EnKF-T (Algorithm 4.2) with $L = 20$ and covariance tapering (SM4.3) with radius $r = 5$. We compare with AD-PF-T (see section SM2) with $L = 20$ and EM (see section SM3). L is chosen from the set $\{1, 5, 10, 20, 50, 100\}$ with the lowest forecast RMSE on the test set at the final training iteration. The implementation details, including the choice of learning rates and other hyperparameters, are discussed in section SM4.

Comparison of the three algorithms is shown in Figure 6. Our AD-EnKF-T recovers α^* better than the other two approaches. The EM approach converges faster, but has a larger error. Moreover, EM tends to converge to a higher level of learned model error σ_β (defined in (SM4.4)), while our AD-EnKF-T shows a consistent drop of learned error level. Note

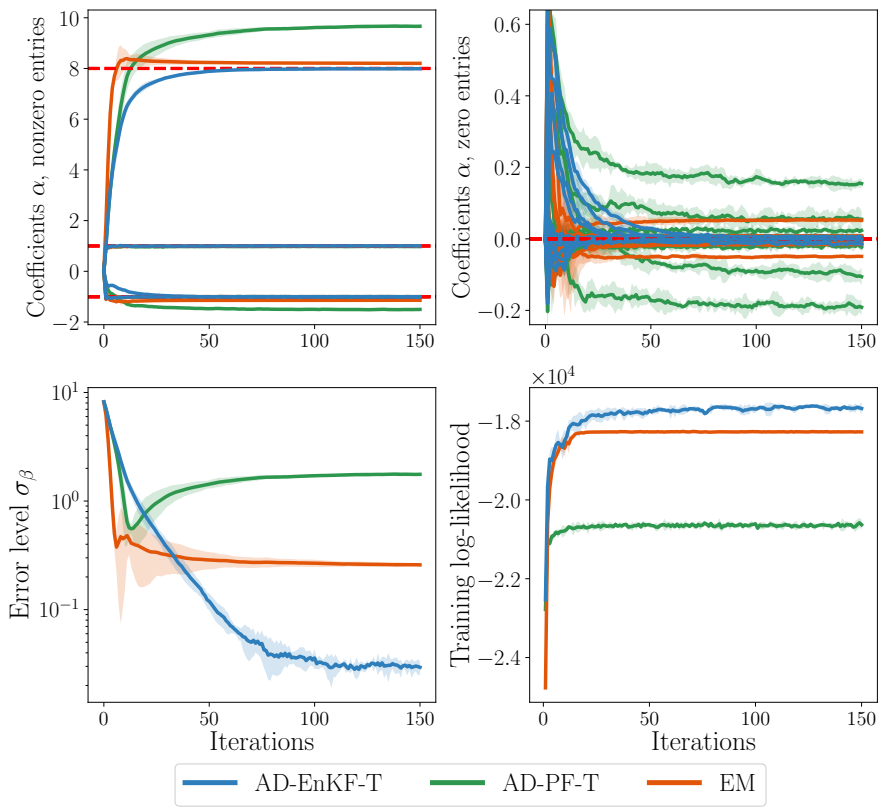


Figure 6. *Learning parameterized dynamics of Lorenz-96 (5.3), with $d_x = 40$ and $H = I_{40}$. Learned value of the 18 coefficients of α (upper left for nonzero entries and upper right for zero entries, where the truth α^* is plotted in red dashed lines), averaged diagnosed error level σ_β (SM4.4) (lower left), and log-likelihood $\mathcal{L}_{EnKF}/\mathcal{L}_{PF}$ during training (lower right), as a function of training iterations. Throughout, the shaded area corresponds to ±2 std over five repeated runs. (subsubsection 5.2.1).*

that Q_β in the learned transition kernel acts like covariance inflation, which is discussed in subsection 4.3, but is "learned" to be adaptive to the training data rather than manually tuned; therefore, having a nonzero error level σ_β may still be helpful. The plot of the log-likelihood estimate during training indicates that AD-EnKF-T searches for parameters with a higher log-likelihood than the EM approach, which is not surprising as AD-EnKF-T directly optimizes $\mathcal{L}_{\text{EnKF}}$, while EM does so by alternatively optimizing a surrogate objective. Also, the large discrepancy between the optimized $\mathcal{L}_{\text{EnKF}}$ and $\mathcal{L}_{\text{PF}}$ objective may be due to $\mathcal{L}_{\text{PF}}$ being a worse estimate for the true log-likelihood $\mathcal{L}$ than that of $\mathcal{L}_{\text{EnKF}}$. Note that PFs may not be suitable for high-dimensional systems like the Lorenz-96 model. Even with knowledge of the true reference model and a large number of particles, the PF is not able to capture the filtering distribution well due to the high dimensionality—see, e.g., Figure 5 of [10]. The plots of forecast error, filter error, and test log-likelihood are presented in Figure 7.

We also consider varying the state dimension d_x and observation model H. (The parameterization in (5.3) is valid for any choice of d_x.) We measure the Euclidean distance between the value of learned α at the final training iteration (at convergence) to α^*. The training procedure is repeated five times and the results are shown in Table 5.2. We vary $d_x \in \{10, 20, 40, 80\}$ and consider two settings for H: fully observed at all coordinates, i.e., $H = I_{d_x}$, and partially observed at every two out of three coordinates [81], i.e., $H = [e_1, e_2, e_4, e_5, e_7, \ldots]^\top$, where

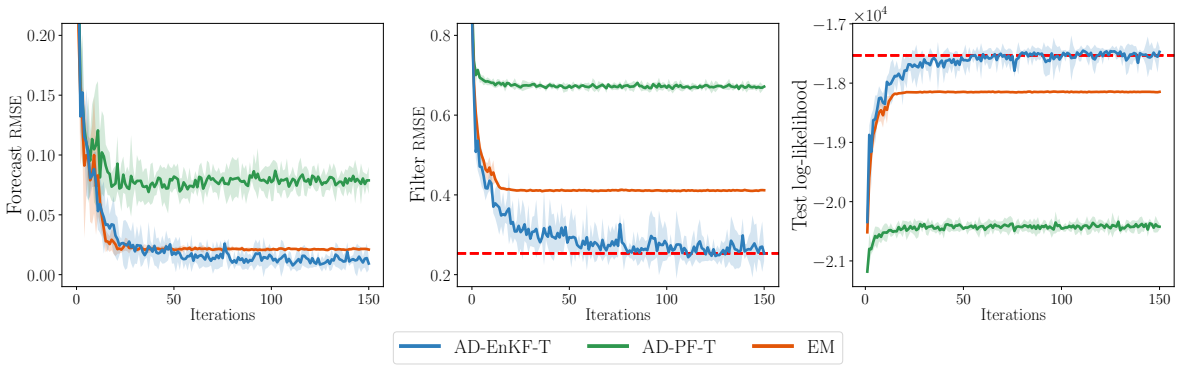


Figure 7. *Learning parameterized dynamics of Lorenz-96 (5.3), with $d_x = 40$ and $H = I_{40}$. All performance metrics are evaluated after each training iteration. Red dashed lines correspond to metric values obtained with the reference model f^* and Q^*. Our proposed AD-EnKF-T performs the best in all metrics, with a performance similar to the reference model.*

Table 5.2

Lorenz-96, learning parameterized dynamics with varying d_x and observation models. The table shows recovery of learned α^* for each algorithm at the final training iteration, in terms of its distance to the truth α^* (5.4). "Full" corresponds to full observations, i.e., $H = I_{d_x}$. "Partial" corresponds to observing two out of three coordinates, i.e., $H = [e_1, e_2, e_4, e_5, e_7, \ldots]^\top$. The "–" indicates that training cannot be completed due to filter divergence (subsubsection 5.2.1).

	$d_x = 10$ (full)	$d_x = 20$ (full)	$d_x = 20$ (partial)	$d_x = 40$ (full)	$d_x = 40$ (partial)	$d_x = 80$ (full)	$d_x = 80$ (partial)
EM	0.308± 0.026	0.289± 0.0114	2.28± 4.92	0.268± 0.0103	7.754± 8.057	0.231± 0.0209	7.382± 4.812
AD-PF-T	0.262± 0.020	0.711± 0.0291	–	1.557± 0.0422	–	2.079± 0.0275	–
AD-EnKF-T	**0.217± 0.027**	**0.0325± 0.0128**	**0.0835± 0.0189**	**0.0283± 0.0022**	**0.0930± 0.0098**	**0.0540± 0.0065**	**0.0813± 0.0083**

$\{e_i\}_{i=1}^{d_x}$ is the standard basis for $\mathbb{R}^{d_x}$. The number of particles used for all algorithms is fixed at $N = 50$, and covariance tapering (SM4.3) with radius $r = 5$ is applied to the EnKF. For both AD-EnKF-T and AD-PF-T, $L = 20$. We find that AD-EnKF-T is able to consistently recover α^* regardless of the choice of d_x and H and is able to perform well in the important case where $N < d_x$, with an accuracy that is orders of magnitude better than the other two approaches. The EM approach is able to recover α^* consistently in fully observed settings, but with a lower accuracy. In partially observed settings, EM does not converge to the same value in repeated runs, possibly due to the existence of multiple local maxima. AD-PF-T is able to converge consistently in fully observed settings but with the lowest accuracy, and runs into filter divergence issues in partially observed settings, so that the training process is not able to complete. Moreover, we observe that the error of AD-PF-T tends to grow with the state dimension d_x, while the two approaches based on EnKF do not deteriorate when increasing the state dimension. This is further evidence that EnKF is superior in high-dimensional settings.

5.2.2. Fully unknown dynamics. We assume no knowledge of the reference vector field f^*, and we approximate it by an NN surrogate, $f_\alpha^{\mathrm{NN}} : \mathbb{R}^{d_x} \to \mathbb{R}^{d_x}$, where α represents the NN weights. The structure of the NN is similar to the one in [12] and is detailed in section SM4. The number of parameters combined for α and β is $d_\theta = 9317$. The experimental results are compared to the model correction results, and hence are postponed to subsubsection 5.2.3.

5.2.3. Model correction. We assume f^* is unknown but that an inaccurate model f_{approx} is available. We make use of the parametric form (5.3) and define f_{approx} via a perturbation $\widetilde{\alpha}$ of the true parameter α^*:

$$f_{\mathrm{approx}} := f_{\widetilde{\alpha}}, \qquad \text{where } \widetilde{\alpha}_i \sim \begin{cases} \mathcal{N}(\alpha_i^*, 1) & \text{if } i = 0, \\ \mathcal{N}(\alpha_i^*, 0.1) & \text{if } i \in \{1, \dots, 5\}, \\ \mathcal{N}(\alpha_i^*, 0.01) & \text{if } i \in \{6, \dots, 17\}. \end{cases} \tag{5.5}$$

The coefficients of a higher-order polynomial have a smaller amount of perturbation. $\widetilde{\alpha}$ is *fixed* throughout the learning procedure. We approximate the residual $f^* - f_{\mathrm{approx}}$ by an NN g_α^{NN}, where α represents the weights, and g_α^{NN} has the same structure and the same number of parameters as in the fully unknown setting. The goal is to learn α so that $f_\alpha := f_{\mathrm{approx}} + g_\alpha^{\mathrm{NN}}$ approximates f^*.

We set $d_x = 40$ and consider two settings for H: fully observed with $H = I_{40}$, $d_y = 40$, and partially observed at every two out of three coordinates with $d_y = 27$ (see subsubsection 5.2.1). Eight data sequences are generated with the reference model for training and four for testing, each with length $T = 1200$. Other experimental settings are the same as in subsubsection 5.2.1.

For the setting where training data is fully observed, we compare AD-EnKF-T with AD-PF-T and the EM approach. The results are plotted in Figure 8. The number of particles used for all algorithms is fixed at $N = 50$, and covariance tapering (SM4.3) with radius $r = 5$ is applied to EnKF. The subsequence length for both AD-EnKF-T and AD-PF-T is chosen to be $L = 20$. We find that, whether f^* is fully known or an inaccurate model is available, AD-EnKF-T is able to learn the reference vector field f^* well, with the smallest forecast

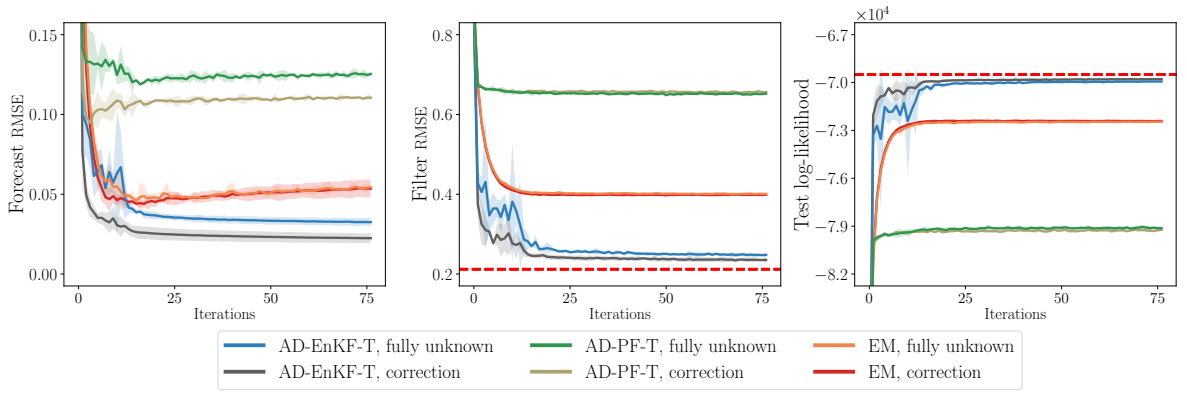


Figure 8. *Learning the Lorenz-96 model from fully unknown dynamics (subsubsection 5.2.2) versus model correction (subsubsection 5.2.3), with full observations ($H = I_{d_x}$). All performance metrics are evaluated after each training iteration. Red dashed lines correspond to metric values obtained with the reference model f^* and Q^*. Our proposed AD-EnKF-T performs the best in all metrics, with a performance similar to the reference model.*

RMSE among all methods. Applying a filtering algorithm to the learned model, we find that the states recovered by the AD-EnKF-T algorithm at the final iteration have the lowest error (filter RMSE) among all methods, indicating that AD-EnKF-T also has the ability to learn unknown states well. Moreover, the filter RMSE of AD-EnKF-T is close to the one computed using a filtering algorithm *with* known f^* and Q^*. The test log-likelihood $\mathcal{L}_{\text{EnKF}}$ of the model learned by AD-EnKF-T is close to the one evaluated with the reference model. We also find that having an inaccurate model f_{approx} is beneficial to the learning of AD-EnKF-T. The performance metrics are boosted compared to the ones with a fully unknown model, in agreement with [59]. EM has worse results, where we find that the forecast RMSE does not consistently drop in the training procedure and the states are not accurately recovered. This might be because the smoothing distribution used by EM cannot be approximated accurately. AD-PF-T has the worst performance, possibly because PF fails in high dimensions.

We repeat the learning procedure in the setting where training data is partially observed at every two out of three coordinates. The results are shown in Figure 9. Those for AD-PF-T are not shown since training cannot be completed due to filter divergence. We find that AD-EnKF-T is still able to recover f^* consistently as well as the unknown states for all coordinates, including the ones that are not observed, and has a filter RMSE close to the one computed with knowledge of f^*. However, the performance metrics of the EM algorithm in the model correction experiment deteriorate as training proceeds, indicating that it may overfit the training data. In addition, we find that the EM algorithm does not converge to the same point in repeated trials, particularly so in the setting of fully unknown dynamics. All of these results indicate that AD-EnKF is advantageous when learning from partial observations in high dimensions.

The ability to recover the underlying dynamics and states even with incomplete observations and fully unknown dynamics is most likely due to the convolutional-type architecture of the NN f_α^{NN}, which implicitly assumes that each coordinate only interacts with its neighbors, and that this interaction is spatially invariant.

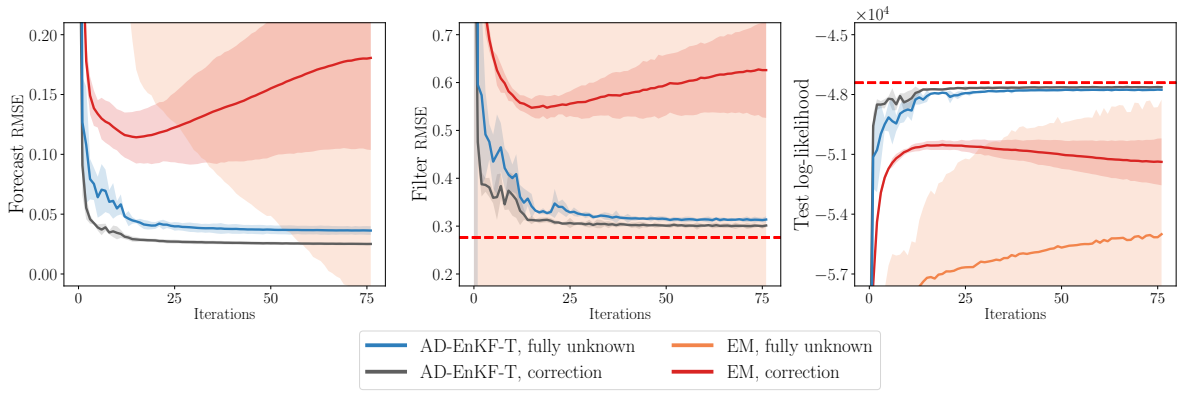


Figure 9. *Learning Lorenz-96 from fully unknown dynamics (subsubsection 5.2.2) versus model correction (subsubsection 5.2.3) with partial observations ($H = [e_1, e_2, e_4, e_5, e_7, \ldots]^\top$). All performance metrics are evaluated after each training iteration. Red dashed lines correspond to metric values obtained with the reference model f^* and Q^*. The absence of lines for EM in the fully unknown setting is due to its low and unstable performance. When compared to the EM method, our proposed AD-EnKF-T is more stable during training and performs better in all metrics, and its performance is closer to the one achieved by the reference model.*

6. Conclusions and future directions. This paper introduced AD-EnKFs for the principled learning of states and dynamics in DA. We have shown that AD-EnKFs can be successfully integrated with DA localization techniques for recovery of high-dimensional states, and with TBPTT techniques to handle large observation data and high-dimensional surrogate models. Numerical results on the Lorenz-96 model show that AD-EnKFs outperform existing EM and PF methods to merge DA and ML.

Several research directions stem from this work. First, gradient and Hessian information of $\mathcal{L}_{\mathrm{EnKF}}$ obtained by autodiff can be utilized to design optimization schemes beyond the first-order approach we consider. Second, the convergence analysis of EnKF estimation of the log-likelihood and its gradient may be generalized to nonlinear settings. It would also be interesting to derive a dimension-dependent bound for the L^p estimation error and the bias $\left|\mathbb{E}\,\mathcal{L}_{\mathrm{EnKF}} - \mathcal{L}\right|$. Third, the idea of AD-EnKF could be applied to autodifferentiate through other filtering algorithms, e.g., unscented Kalman filters, and in Bayesian inverse problems using iterative ensemble Kalman methods. The paper [49] is an important first step in this direction. Replacing EnKF analysis steps with differentiable optimal transport maps [18] is also a promising future direction. Finally, the encouraging numerical results obtained on the Lorenz-96 model motivate the deployment and further investigation of AD-EnKFs in scientific and engineering applications where latent states need to be estimated with incomplete knowledge of their dynamics.

Appendix A. Proof of Theorem 3.1.

Notation. We denote by c a constant that does not depend on N and may change from line to line. We denote by $\|U\|_p$ the L^p norm of a random vector/matrix U: $\|U\|_p := \left(\mathbb{E}\,|U|^p\right)^{1/p}$, where $|\cdot|$ is the underlying vector/matrix norm. (Here we use 2-norm for vectors and Frobenius norm for matrices.) For a sequence of random vectors/matrices U_N, we write

$$U_N \xrightarrow{1/2} U$$

if, for any $p \geq 1$, there exists a constant c such that

$$\|U_N - U\|_p \leq cN^{-1/2} \qquad \forall N \geq 1.$$

For a scalar valued function $f(U)$ that takes a vector/matrix U as input, we denote by $\partial_U f$ the derivative of f w.r.t. U, which collects the derivative of f w.r.t. each entry of the vector/matrix U. When U is a vector, the notations $\partial_U f$ and $\nabla_U f$ are equivalent. For a vector/matrix valued function $U(a)$ that takes a scalar a as input, we denote by $\partial_a U$ the derivative of U w.r.t. a, which collects the derivative of each entry of the vector/matrix U w.r.t. a.

We first recall the propagation of chaos statement. Notice that in the EnKF algorithm, Algorithm 3.1, we compute $x_t^{1:N}$ sequentially, based on the forecast ensemble $\widehat{x}_t^{1:N}$ and its empirical mean and covariance $\widehat{m}_t, \widehat{C}_t$. We build "substitute particles" $\mathbf{x}_t^{1:N}$ in a similar fashion, except that at each step the population mean and covariance $\widehat{\mathbf{m}}_t, \widehat{\mathbf{C}}_t$ are used instead of their empirical versions. Starting from $\mathbf{x}_0^{1:N} = x_0^{1:N}$, the update rules of substitute particles are listed below, with a side-by-side comparison to the EnKF update rules:

(A.1)

EnKF particles		Substitute particles	
$\widehat{x}_t^n$	$= F_\alpha(x_{t-1}^n) + \xi_t^n$	$\widehat{\mathbf{x}}_t^n$	$= F_\alpha(\mathbf{x}_{t-1}^n) + \xi_t^n$
$\widehat{m}_t$	$= \frac{1}{N}\sum_{n=1}^N \widehat{x}_t^n$	$\widehat{\mathbf{m}}_t$	$= \mathbb{E}[\widehat{\mathbf{x}}_t^n]$
$\widehat{C}_t$	$= \frac{1}{N-1}\sum_{n=1}^N (\widehat{x}_t^n - \widehat{m}_t)(\widehat{x}_t^n - \widehat{m}_t)^\top$	$\widehat{\mathbf{C}}_t$	$= \mathbb{E}[(\widehat{\mathbf{x}}_t^n - \widehat{\mathbf{m}}_t)(\widehat{\mathbf{x}}_t^n - \widehat{\mathbf{m}}_t)^\top]$
$\widehat{K}_t$	$= \widehat{C}_t H^\top (H\widehat{C}_t H^\top + R)^{-1}$	$\widehat{\mathbf{K}}_t$	$= \widehat{\mathbf{C}}_t H^\top (H\widehat{\mathbf{C}}_t H^\top + R)^{-1}$
x_t^n	$= \widehat{x}_t^n + \widehat{K}_t(y_t + \gamma_t^n - H\widehat{x}_t^n)$	$\mathbf{x}_t^n$	$= \widehat{\mathbf{x}}_t^n + \widehat{\mathbf{K}}_t(y_t + \gamma_t^n - H\widehat{\mathbf{x}}_t^n)$

Notice that the substitute particles use the *same* realization of random variables as the EnKF particles, including initialization of particles $x_0^{1:N}$, forecast simulation error ξ_t^n, and noise perturbation γ_t^n. As $N \to \infty$, one can show that the EnKF particles $x_t^{1:N}$ (resp., $\widehat{x}_t^{1:N}$) are close to the substitute particles $\mathbf{x}_t^{1:N}$ (resp., $\widehat{\mathbf{x}}_t^{1:N}$), and hence the law of large numbers guarantees that $\widehat{m}_t, \widehat{C}_t$ are close to $\widehat{\mathbf{m}}_t, \widehat{\mathbf{C}}_t$. We summarize the main results from [58] (see also [51]).

Lemma A.1. *Under the same assumption of Theorem 3.1, we have the following:*
(1) *For each $t \geq 1$, the substitute particles $\mathbf{x}_t^{1:N}$ are i.i.d., and each of them has the same law as the true filtering distribution $p(x_t|y_{1:t})$. Similarly, $\widehat{\mathbf{x}}_t^{1:N}$ are i.i.d., and each of them has the same law as the true forecast distribution $p(x_t|y_{1:t-1})$. In particular,*

$$p(x_t|y_{1:t-1}) = \mathcal{N}(x_t; \widehat{\mathbf{m}}_t, \widehat{\mathbf{C}}_t). \tag{A.2}$$

(2) *For each $t, n, p \geq 1$, the EnKF particle x_t^n converges to the substitute particle $\mathbf{x}_t^n$ in L^p with convergence rate $N^{-1/2}$, and the substitute particle $\mathbf{x}_t^n$ has finite moments of any order. The same holds for forecast particles $\widehat{x}_t^n$:*

$$x_t^n \xrightarrow{1/2} \mathbf{x}_t^n, \qquad \widehat{x}_t^n \xrightarrow{1/2} \widehat{\mathbf{x}}_t^n, \qquad \|\mathbf{x}_t^n\|_p \leq c, \qquad \|\widehat{\mathbf{x}}_t^n\|_p \leq c. \tag{A.3}$$

In particular, $\widehat{m}_t$, $\widehat{C}_t$ converge to $\widehat{\mathbf{m}}_t$, $\widehat{\mathbf{C}}_t$ in L^p with convergence rate $N^{-1/2}$:

$$\widehat{m}_t \xrightarrow{1/2} \widehat{\mathbf{m}}_t, \qquad \widehat{C}_t \xrightarrow{1/2} \widehat{\mathbf{C}}_t. \tag{A.4}$$

Proof. (A.2) corresponds to Lemma 2.1 of [58]. (A.3) corresponds to Proposition 4.4 of [58]. (A.4) is a direct corollary of Theorem 5.2 of [58]. ∎

Proof of Theorem 3.1. By (A.2), using the Gaussian observation assumption (2.2),

$$\mathcal{L}(\theta) = \sum_{t=1}^{T} \log p(y_t|y_{1:t-1}) = \sum_{t=1}^{T} \log \mathcal{N}(y_t; H\widehat{\mathbf{m}}_t, H\widehat{\mathbf{C}}_t H^\top + R). \tag{A.5}$$

By (3.6),

$$\mathcal{L}_{\mathrm{EnKF}}(\theta) = \sum_{t=1}^{T} \log \mathcal{N}(y_t; H\widehat{m}_t, H\widehat{C}_t H^\top + R). \tag{A.6}$$

Define

$$\begin{aligned} h_t(m, C) &:= \log \mathcal{N}(y_t; Hm, HCH^\top + R) \\ &= -\frac{1}{2}\log\det(HCH^\top + R) - \frac{1}{2}(y_t - Hm)^\top (HCH^\top + R)^{-1}(y_t - Hm) + \text{const.} \end{aligned} \tag{A.7}$$

It suffices to show that, for each $t \geq 1$,

$$h_t(\widehat{m}_t, \widehat{C}_t) \xrightarrow{1/2} h_t(\widehat{\mathbf{m}}_t, \widehat{\mathbf{C}}_t). \tag{A.8}$$

We denote by $\mathbb{S}_+^{d_x} \subset \mathbb{R}^{d_x \times d_x}$ the space of all positive semidefinite matrices equipped with Frobenius norm. Notice that h_t is a continuous function on $\mathbb{R}^{d_x} \times \mathbb{S}_+^{d_x}$, since $HCH^\top + R \succeq R \succ 0$. To show convergence in L^p, intuitively one would expect a Lipschitz-type continuity to hold for h_t, in a suitable sense. We inspect the derivatives of h_t w.r.t. m and C, which will also be useful for later developments:

$$\begin{aligned} \partial_m h_t(m, C) &= -H^\top (HCH^\top + R)^{-1}(y_t - Hm), \\ \partial_C h_t(m, C) &= -\frac{1}{2} H^\top (HCH^\top + R)^{-1} H \\ &\quad + \frac{1}{2} H^\top (HCH^\top + R)^{-1}(y_t - Hm)(y_t - Hm)^\top (HCH^\top + R)^{-1} H. \end{aligned} \tag{A.9}$$

Since $R^{d_x} \times \mathbb{S}_+^{d_x}$ is convex, by the mean value theorem, triangle inequality, and Cauchy–Schwarz, and define $m(\chi) := \chi\widehat{\mathbf{m}}_t + (1-\chi)\widehat{m}_t$, $C(\chi) := \chi\widehat{\mathbf{C}}_t + (1-\chi)\widehat{C}_t$,

$$\begin{aligned} \left|h_t(\widehat{m}_t, \widehat{C}_t) - h_t(\widehat{\mathbf{m}}_t, \widehat{\mathbf{C}}_t)\right| &\leq \sup_{\chi\in[0,1]} \left|\partial_m h_t\big(m(\chi), C(\chi)\big)\right| |\widehat{m}_t - \widehat{\mathbf{m}}_t| \\ &\quad + \sup_{\chi\in[0,1]} \left|\partial_C h_t\big(m(\chi), C(\chi)\big)\right| |\widehat{C}_t - \widehat{\mathbf{C}}_t|. \end{aligned} \tag{A.10}$$

Taking L_p norm on both sides,

$$\begin{aligned} \left\|h_t(\widehat{m}_t, \widehat{C}_t) - h_t(\widehat{\mathbf{m}}_t, \widehat{\mathbf{C}}_t)\right\|_p &\leq \sup_{\chi\in[0,1]} \left\|\partial_m h_t\big(m(\chi), C(\chi)\big)\right\|_{2p} \|\widehat{m}_t - \widehat{\mathbf{m}}_t\|_{2p} \\ &\quad + \sup_{\chi\in[0,1]} \left\|\partial_C h_t\big(m(\chi), C(\chi)\big)\right\|_{2p} \|\widehat{C}_t - \widehat{\mathbf{C}}_t\|_{2p}, \end{aligned} \tag{A.11}$$

where we have used the triangle inequality and the L^p Cauchy–Schwarz inequality $\|\|U||V|\|_p \le \|U\|_{2p}\|V\|_{2p}$; see, e.g., Lemma 2.1 of [51]. Also, by plugging in (A.9), for each $\chi \in [0,1]$,

$$\begin{aligned}\big\|\partial_m h_t\big(m(\chi),C(\chi)\big)\big\|_{2p} &\le \big\||H||(HC(\chi)H^\top + R)^{-1}|(|y_t - \chi H\widehat{\mathbf{m}}_t - (1-\chi)H\widehat{m}_t|)\big\|_{2p}\\ &\le |H||R^{-1}|\big(|y_t| + |H||\widehat{\mathbf{m}}_t| + |H|\|\widehat{m}_t\|_{2p}\big) \le c,\end{aligned} \tag{A.12}$$

where we have used that $|(HC(\chi)H^\top + R)^{-1}| \le |R^{-1}|$, that $\widehat{\mathbf{m}}_t$ is deterministic, and that all moments of $\widehat{m}_t$ are finite, by (A.4). Similarly,

$$\begin{aligned}\big\|\partial_C h_t\big(m(\chi),C(\chi)\big)\big\|_{2p} &\le \frac{1}{2}|H|^2|R^{-1}| + \frac{1}{2}|H|^2|R^{-1}|^2\|y_t - \chi H\widehat{\mathbf{m}}_t - (1-\chi)H\widehat{m}_t\|_{4p}^2\\ &\le \frac{1}{2}|H|^2|R^{-1}| + \frac{1}{2}|H|^2|R^{-1}|^2(|y_t| + |H||\widehat{\mathbf{m}}_t| + |H|\|\widehat{m}_t\|_{4p})^2 \le c,\end{aligned} \tag{A.13}$$

where we have used that $|vv^\top| = |v|^2$ for vector v. Thus, combining (A.4) and (A.11)–(A.13) gives

$$\|h_t(\widehat{m}_t,\widehat{C}_t) - h_t(\widehat{\mathbf{m}}_t,\widehat{\mathbf{C}}_t)\|_p \le cN^{-1/2}, \tag{A.14}$$

which concludes the proof. ∎

Appendix B. Proof of Theorem 3.2. Without loss of generality, we assume that $\theta \in \mathbb{R}$ is a scalar parameter, since in general the gradient w.r.t. θ is a collection of derivatives w.r.t. each element of θ. We will use the following lemma repeatedly.

Lemma B.1. *For sequences of random vectors/matrices U_N, V_N,*

(1) *If $U_N - V_N \xrightarrow{1/2} 0$ and $V_N \xrightarrow{1/2} V$, then*

$$U_N \xrightarrow{1/2} V. \tag{B.1}$$

(2) *If $U_N \xrightarrow{1/2} U$, $V_N \xrightarrow{1/2} V$, and U, V have finite moments of any order, then*

$$U_N V_N \xrightarrow{1/2} UV. \tag{B.2}$$

More generally, the result holds for multiplication of more than two variables.

Proof. (1) Using $U_N - V = (U_N - V_N) + (V_N - V)$, the proof follows from the triangle inequality.

(2) Applying the triangle inequality and the L^p Cauchy–Schwarz inequality,

$$\begin{aligned}\|U_N V_N - UV\|_p &\le \|(U_N - U)V_N\|_p + \|U(V_N - V)\|_p\\ &\le \|U_N - U\|_{2p}\|V_N\|_{2p} + \|U\|_{2p}\|V_N - V\|_{2p}\\ &\le cN^{-1/2}\Big(\|V\|_{2p} + cN^{-1/2}\Big) + \|U\|_{2p}cN^{-1/2}\\ &\le cN^{-1/2}.\end{aligned} \tag{B.3}$$

∎

The following result, which we will use repeatedly, is an immediate corollary of Lemma A.1.

Lemma B.2. *Under the same assumption as Theorem 3.1,*

$$(H\widehat{C}_t H^\top + R)^{-1} \xrightarrow{1/2} (H\widehat{\mathbf{C}}_t H^\top + R)^{-1}. \tag{B.4}$$

Proof. Using the identity $A^{-1} - B^{-1} = A^{-1}(B-A)B^{-1}$ for invertible matrices A, B,

$$\begin{aligned} &\|(H\widehat{C}_t H^\top + R)^{-1} - (H\widehat{\mathbf{C}}_t H^\top + R)^{-1}\|_p \\ &= \|(H\widehat{C}_t H^\top + R)^{-1} H(\widehat{\mathbf{C}}_t - \widehat{C}_t)H^\top (H\widehat{\mathbf{C}}_t H^\top + R)^{-1}\|_p \\ &\le |R^{-1}|^2 |H|^2 \|\widehat{C}_t - \widehat{\mathbf{C}}_t\|_p \\ &\le cN^{-1/2}, \end{aligned} \tag{B.5}$$

where we have used the L^p convergence of $\widehat{C}_t$ to $\widehat{\mathbf{C}}_t$ (A.4) and the fact that $|(HCH^T + R)^{-1}| \le |R^{-1}|$ for $C \succeq 0$. ∎

Lemma B.3. *Under the same assumption as Theorem 3.2, for each $t \ge 1$, both $\partial_\theta \widehat{x}_t^n$ and $\partial_\theta \widehat{\mathbf{x}}_t^n$ exist, and $\partial_\theta \widehat{x}_t^n$ converges to $\partial_\theta \widehat{\mathbf{x}}_t^n$ in L^p for any $p \ge 1$ with convergence rate $N^{-1/2}$. Moreover, $\partial_\theta \widehat{\mathbf{x}}_t^n$ has finite moments of any order:*

$$\partial_\theta \widehat{x}_t^n \xrightarrow{1/2} \partial_\theta \widehat{\mathbf{x}}_t^n, \qquad \|\partial_\theta \widehat{\mathbf{x}}_t^n\|_p \le c \qquad \forall n. \tag{B.6}$$

In addition, all derivatives $\partial_\theta \widehat{m}_t$, $\partial_\theta \widehat{\mathbf{m}}_t$, $\partial_\theta \widehat{C}_t$, $\partial_\theta \widehat{\mathbf{C}}_t$, $\partial_\theta \widehat{K}_t$, and $\partial_\theta \widehat{\mathbf{K}}_t$ exist, and

$$\partial_\theta \widehat{m}_t \xrightarrow{1/2} \partial_\theta \widehat{\mathbf{m}}_t, \qquad \partial_\theta \widehat{C}_t \xrightarrow{1/2} \partial_\theta \widehat{\mathbf{C}}_t, \qquad \partial_\theta \widehat{K}_t \xrightarrow{1/2} \partial_\theta \widehat{\mathbf{K}}_t. \tag{B.7}$$

Proof. We will prove this by induction. For $t = 1$, since $\widehat{x}_1^n = Ax_0^n + S\xi_0^n = \widehat{\mathbf{x}}_1^n$,

$$\partial_\theta \widehat{x}_1^n = (\partial_\theta A)x_0^n + (\partial_\theta S)\xi_0^n = \partial_\theta \widehat{\mathbf{x}}_1^n, \tag{B.8}$$

and both derivatives $\partial_\theta \widehat{x}_1^n$ and $\partial_\theta \widehat{\mathbf{x}}_1^n$ exist. Also,

$$\|\partial_\theta \widehat{\mathbf{x}}_1^n\|_p \le |\partial_\theta A| \|x_0^n\|_p + |\partial_\theta S| \|\xi_0^n\|_p \le c, \tag{B.9}$$

since x_0^n and ξ_0^n are drawn from Gaussian distributions, which have finite moments of any order. So (B.6) holds for $t = 1$.

Assume (B.6) holds for step t. Then, using the definition for $\widehat{m}_t$,

$$\partial_\theta \widehat{m}_t = \frac{1}{N}\sum_{n=1}^N \partial_\theta \widehat{x}_t^n \underset{①}{\xrightarrow{1/2}} \frac{1}{N}\sum_{n=1}^N \partial_\theta \widehat{\mathbf{x}}_t^n \underset{②}{\xrightarrow{1/2}} \mathbb{E}[\partial_\theta \widehat{\mathbf{x}}_t^n] \underset{③}{=} \partial_\theta \mathbb{E}[\widehat{x}_t^n] = \partial_\theta \widehat{\mathbf{m}}_t. \tag{B.10}$$

Convergence ① follows from induction assumption (B.6). Convergence ② follows from law of large numbers in L^p, since $\partial_\theta \widehat{\mathbf{x}}_t^n$ are i.i.d. and the moments of $\partial_\theta \widehat{\mathbf{x}}_t^n$ are finite by induction assumption (B.6). The swap of differentiation and expectation in ③ is valid since the expectation is taken over a distribution that is independent of θ. Both derivatives $\partial_\theta \widehat{m}_t$ and $\partial_\theta \widehat{\mathbf{m}}_t$ exist. Similarly,

(B.11)

$$
\begin{aligned}
\partial_\theta \widehat{C}_t &= \frac{1}{N-1}\sum_{n=1}^{N} \partial_\theta(\widehat{x}_t^n(\widehat{x}_t^n)^\top) - \frac{N}{N-1}\partial_\theta(\widehat{m}_t\widehat{m}_t^\top) \\
&= \frac{1}{N-1}\sum_{n=1}^{N} \big((\partial_\theta \widehat{x}_t^n)(\widehat{x}_t^n)^\top + \widehat{x}_t^n(\partial_\theta \widehat{x}_t^n)^\top\big) - \frac{N}{N-1}\big((\partial_\theta \widehat{m}_t)\widehat{m}_t^\top + \widehat{m}_t(\partial_\theta \widehat{m}_t)^\top\big) \\
&\xrightarrow[①]{1/2} \frac{1}{N-1}\sum_{n=1}^{N} \big((\partial_\theta \widehat{\mathbf{x}}_t^n)(\widehat{\mathbf{x}}_t^n)^\top + \widehat{\mathbf{x}}_t^n(\partial_\theta \widehat{\mathbf{x}}_t^n)^\top\big) - \frac{N}{N-1}\big((\partial_\theta \widehat{\mathbf{m}}_t)\widehat{\mathbf{m}}_t^\top + \widehat{\mathbf{m}}_t(\partial_\theta \widehat{\mathbf{m}}_t)^\top\big) \\
&\xrightarrow[②]{1/2} \mathbb{E}[\partial_\theta\big(\widehat{\mathbf{x}}_t^n(\widehat{\mathbf{x}}_t^n\big)^\top)] - \partial_\theta(\widehat{\mathbf{m}}_t\widehat{\mathbf{m}}_t^\top) \\
&\underset{③}{=} \partial_\theta(\mathbb{E}[\widehat{\mathbf{x}}_t^n(\widehat{\mathbf{x}}_t^n)^\top] - \widehat{\mathbf{m}}_t\widehat{\mathbf{m}}_t^\top) \\
&= \partial_\theta \widehat{\mathbf{C}}_t.
\end{aligned}
$$

For ① we have used the L^p convergence of $\widehat{x}_t^n$ to $\widehat{\mathbf{x}}_t^n$, $\partial_\theta \widehat{x}_t^n$ to $\partial_\theta \widehat{\mathbf{x}}_t^n$, $\widehat{m}_t$ to $\widehat{\mathbf{m}}_t$, and $\partial_\theta \widehat{m}_t$ to $\partial_\theta \widehat{\mathbf{m}}_t$ with rate $N^{-1/2}$ and the fact that $\widehat{\mathbf{x}}_t^n$ and $\partial_\theta \widehat{\mathbf{x}}_t^n$ have finite moments of any order, followed by Lemma B.1. Convergence ② follows from law of large numbers in L^p since $(\partial_\theta \widehat{\mathbf{x}}_t^n)(\widehat{\mathbf{x}}_t^n)^\top$ are i.i.d. with finite moments, by the Cauchy–Schwarz inequality. ③ is valid since the expectation is taken over a distribution that is independent of θ. Both derivatives $\partial_\theta \widehat{C}_t$ and $\partial_\theta \widehat{\mathbf{C}}_t$ exist. Similarly,

(B.12)

$$
\begin{aligned}
\partial_\theta \widehat{K}_t &= \partial_\theta\big(\widehat{C}_t H(H\widehat{C}_t H^\top + R)^{-1}\big) \\
&\underset{①}{=} (\partial_\theta \widehat{C}_t)H(H\widehat{C}_t H^\top + R)^{-1} \\
&\qquad - \widehat{C}_t H(H\widehat{C}_t H^\top + R)^{-1}(H(\partial_\theta \widehat{C}_t)H^\top + R)(H\widehat{C}_t H^\top + R)^{-1} \\
&\xrightarrow[②]{1/2} (\partial_\theta \widehat{\mathbf{C}}_t)H(H\widehat{\mathbf{C}}_t H^\top + R)^{-1} \\
&\qquad - \widehat{\mathbf{C}}_t H(H\widehat{\mathbf{C}}_t H^\top + R)^{-1}(H(\partial_\theta \widehat{\mathbf{C}}_t)H^\top + R)(H\widehat{\mathbf{C}}_t H^\top + R)^{-1} \\
&\underset{①}{=} \partial_\theta\big(\widehat{\mathbf{C}}_t H(H\widehat{\mathbf{C}}_t H^\top + R)^{-1}\big) \\
&= \partial_\theta \widehat{\mathbf{K}}_t.
\end{aligned}
$$

Here equalities ① and ③ follow from chain rule. For ② we have used the L^p convergence of $\widehat{C}_t$ to $\widehat{\mathbf{C}}_t$, $\partial_\theta \widehat{C}_t$ to $\partial_\theta \widehat{\mathbf{C}}_t$, and $(H\widehat{C}_t H^\top + R)^{-1}$ to $(H\widehat{\mathbf{C}}_t H^\top + R)^{-1}$ with rate $N^{-1/2}$ (by (B.4)), followed by Lemma B.1. Both derivatives $\partial_\theta \widehat{K}_t$ and $\partial_\theta \widehat{\mathbf{K}}_t$ exist since $R \succ 0$.

To show (B.6) holds for step $t+1$, we need to investigate the derivatives of the analysis ensemble $\partial_\theta x_t^n$, by plugging in the EnKF update formula:

(B.13)

$$
\begin{aligned}
\partial_\theta x_t^n &= \partial_\theta(\widehat{x}_t^n + \widehat{K}_t(y_t + \gamma_t^n - H\widehat{x}_t^n) \\
&\underset{①}{=} (I - \widehat{K}_t H)\partial_\theta \widehat{x}_t^n + (\partial_\theta \widehat{K}_t)(y_t + \gamma_t^n - H\widehat{x}_t^n) \\
&= \big(I - \widehat{C}_t H^\top(H\widehat{C}_t H^\top + R)^{-1}H\big)\partial_\theta \widehat{x}_t^n + (\partial_\theta \widehat{K}_t)(y_t + \gamma_t^n - H\widehat{x}_t^n) \\
&\xrightarrow[②]{1/2} (I - \widehat{\mathbf{C}}_t H^\top(H\widehat{\mathbf{C}}_t H^\top + R)^{-1}H)\partial_\theta \widehat{\mathbf{x}}_t^n + (\partial_\theta \widehat{\mathbf{K}}_t)(y_t + \gamma_t^n - H\widehat{\mathbf{x}}_t^n)
\end{aligned}
$$

$$
\begin{aligned}
&= (I - \widehat{\mathbf{K}}_t H)\partial_\theta \widehat{\mathbf{x}}_t^n + (\partial_\theta \widehat{\mathbf{K}}_t)(y_t + \gamma_t^n - H\widehat{\mathbf{x}}_t^n) \\
&\underset{③}{=} \partial_\theta(\widehat{\mathbf{x}}_t^n + \widehat{\mathbf{K}}_t(y_t + \gamma_t^n - H\widehat{\mathbf{x}}_t^n) \\
&= \partial_\theta \mathbf{x}_t^n.
\end{aligned}
$$

Equalities ① and ③ follow from the chain rule. For ② we have used the L^p convergence of $\widehat{x}_t^n$ to $\widehat{\mathbf{x}}_t^n$, $\partial_\theta \widehat{x}_t^n$ to $\partial_\theta \widehat{\mathbf{x}}_t^n$, $\widehat{C}_t$ to $\widehat{\mathbf{C}}_t$, $\partial_\theta \widehat{K}_t$ to $\partial_\theta \widehat{\mathbf{K}}_t$, and $(H\widehat{C}_t H^\top + R)^{-1}$ to $(H\widehat{\mathbf{C}}_t H^\top + R)^{-1}$, with convergence rate $N^{-1/2}$, and the fact that $\widehat{\mathbf{x}}_t^n$, $\partial_\theta \widehat{\mathbf{x}}_t^n$ and the Gaussian random variable γ_t^n have finite moments of any order, followed by Lemma B.1. Both derivatives $\partial_\theta x_t^n$ and $\partial_\theta \mathbf{x}_t^n$ exist since $R \succ 0$. We also have the moment bound on $\partial_\theta \mathbf{x}_t^n$,

$$
\|\partial_\theta \mathbf{x}_t^n\| \le |I - \widehat{\mathbf{K}}_t H| \|\partial_\theta \widehat{\mathbf{x}}_t^n\|_p + |\partial_\theta \widehat{\mathbf{K}}_t|(|y_t| + \|\gamma_t^n\|_p + |H|\|\widehat{\mathbf{x}}_t^n\|_p) \le c, \tag{B.14}
$$

since $\widehat{\mathbf{x}}_t^n$, $\partial_\theta \widehat{\mathbf{x}}_t^n$, and the Gaussian random variable γ_t^n have finite moments of any order. Then,

$$
\begin{aligned}
\partial_\theta \widehat{x}_{t+1}^n &= \partial_\theta(A x_t^n + S\xi_t^n) \\
&\underset{①}{=} (\partial_\theta A) x_t^n + A(\partial_\theta x_t^n) + (\partial_\theta S)\xi_t^n \\
&\xrightarrow[②]{1/2} (\partial_\theta A)\mathbf{x}_t^n + A(\partial_\theta \mathbf{x}_t^n) + (\partial_\theta S)\xi_t^n \\
&\underset{③}{=} \partial_\theta(A\mathbf{x}_t^n + S\xi_t^n) \\
&= \partial_\theta \widehat{\mathbf{x}}_{t+1}^n.
\end{aligned} \tag{B.15}
$$

Here equalities ① and ③ follow from chain rule. For ② we have used the L^p convergence of x_t^n to $\mathbf{x}_t^n$ and $\partial_\theta x_t^n$ to $\partial_\theta \mathbf{x}_t^n$. Both derivatives $\partial_\theta \widehat{x}_{t+1}^n$ and $\partial_\theta \widehat{\mathbf{x}}_{t+1}^n$ exist since both $\partial_\theta x_t^n$ and $\partial_\theta \mathbf{x}_t^n$ exist. We also have the moment bound

$$
\|\partial_\theta \widehat{\mathbf{x}}_{t+1}^n\|_p \le |\partial_\theta A| \|\mathbf{x}_t^n\|_p + |A| \|\partial_\theta \mathbf{x}_t^n\|_p + |\partial_\theta S| \|\xi_t^n\|_p \le c, \tag{B.16}
$$

since $\mathbf{x}_t^n$, $\partial_\theta \mathbf{x}_t^n$, and Gaussian random variable ξ_t^n have finite moments of any order. Thus (B.6) is proved for step $t+1$ and the induction step is finished, which concludes the proof of the lemma. ∎

Proof of Theorem 3.2. Recall the definition of h_t (A.7). It suffices to show that, for each $t \ge 1$,

$$
\partial_\theta \Big(h_t(\widehat{m}_t, \widehat{C}_t) \Big) \xrightarrow{1/2} \partial_\theta \Big(h_t(\widehat{\mathbf{m}}_t, \widehat{\mathbf{C}}_t) \Big). \tag{B.17}
$$

We first investigate the convergence of derivatives of h_t w.r.t. $\widehat{m}_t$ and $\widehat{C}_t$. The derivatives are computed in (A.9):

$$
\begin{aligned}
\partial_m h_t(\widehat{m}_t, \widehat{C}_t) &= -H^\top (H\widehat{C}_t H^\top + R)^{-1}(y_t - H\widehat{m}_t) \\
&\xrightarrow{1/2} -H^\top (H\widehat{\mathbf{C}}_t H^\top + R)^{-1}(y_t - H\widehat{\mathbf{m}}_t) \\
&= \partial_m h_t(\widehat{\mathbf{m}}_t, \widehat{\mathbf{C}}_t),
\end{aligned} \tag{B.18}
$$

and

$$\begin{aligned}\partial_C h_t(\widehat{m}_t, \widehat{C}_t) &= -\frac{1}{2}H^\top (H\widehat{C}_t H^\top + R)^{-1} H \\ &\quad + \frac{1}{2}H^\top (H\widehat{C}_t H^\top + R)^{-1}(y_t - H\widehat{m}_t)(y_t - H\widehat{m}_t)^\top (H\widehat{C}_t H^\top + R)^{-1} H \\ &\xrightarrow{1/2} -\frac{1}{2}H^\top (H\widehat{\mathbf{C}}_t H^\top + R)^{-1} H \\ &\quad + \frac{1}{2}H^\top (H\widehat{\mathbf{C}}_t H^\top + R)^{-1}(y_t - H\widehat{\mathbf{m}}_t)(y_t - H\widehat{\mathbf{m}}_t)^\top (H\widehat{\mathbf{C}}_t H^\top + R)^{-1} H,\end{aligned} \tag{B.19}$$

where we have used the L^p convergence of $\widehat{m}_t$ to $\widehat{\mathbf{m}}_t$ and $(H\widehat{C}_t H^\top + R)^{-1}$ to $(H\widehat{\mathbf{C}}_t H^\top + R)^{-1}$ by (B.4), followed by Lemma B.1. Then, by the chain rule,

$$\begin{aligned}\partial_\theta\Big(h_t(\widehat{m}_t, \widehat{C}_t)\Big) &= \Big(\partial_m h_t(\widehat{m}_t, \widehat{C}_t)\Big)^\top \partial_\theta \widehat{m}_t + \mathrm{Tr}\Big(\Big(\partial_C h_t(\widehat{m}_t, \widehat{C}_t)\Big)^\top \partial_\theta \widehat{C}_t\Big) \\ &\xrightarrow{1/2} \Big(\partial_m h_t(\widehat{\mathbf{m}}_t, \widehat{\mathbf{C}}_t)\Big)^\top \partial_\theta \widehat{\mathbf{m}}_t + \mathrm{Tr}\Big(\Big(\partial_C h_t(\widehat{\mathbf{m}}_t, \widehat{\mathbf{C}}_t)\Big)^\top \partial_\theta \widehat{\mathbf{C}}_t\Big) \\ &= \partial_\theta\Big(h_t(\widehat{\mathbf{m}}_t, \widehat{\mathbf{C}}_t)\Big).\end{aligned} \tag{B.20}$$

Both derivatives exist since $\partial_\theta \widehat{m}_t$, $\partial_\theta \widehat{\mathbf{m}}_t$, $\partial_\theta \widehat{C}_t$, and $\partial_\theta \widehat{\mathbf{C}}_t$ exist, by Lemma B.3. We have used (B.18) and (B.19) above, the L^p convergence of $\partial_\theta \widehat{m}_t$ to $\partial_\theta \widehat{\mathbf{m}}_t$, and $\partial_\theta \widehat{C}_t$ to $\partial_\theta \widehat{\mathbf{C}}_t$ with rate $N^{-1/2}$ by Lemma B.3, followed by Lemma B.1. ∎

Remark B.4. We again emphasize that all the derivatives and chain rule formulas do *not* need to be computed by hand in applications, but rather through the modern autodiff libraries. We list them out only for the purpose of proving convergence results.

REFERENCES

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., *Tensorflow: A system for large-scale machine learning*, in Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, 2016, pp. 265–283.

[2] S. Agapiou, O. Papaspiliopoulos, D. Sanz-Alonso, and A. M. Stuart, *Importance sampling: Intrinsic dimension and computational cost*, Statist. Sci., 32 (2017), pp. 405–431.

[3] J. L. Anderson, *An ensemble adjustment Kalman filter for data assimilation*, Monthly Weather Review, 129 (2001), pp. 2884–2903.

[4] J. L. Anderson and S. L. Anderson, *A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts*, Monthly Weather Review, 127 (1999), pp. 2741–2758.

[5] C. Andrieu, A. Doucet, and R. Holenstein, *Particle Markov chain Monte Carlo methods*, J. R. Stat. Soc. Ser. B Stat. Methodol., 72 (2010), pp. 269–342.

[6] I. Ayed, E. de Bézenac, A. Pajot, J. Brajard, and P. Gallinari, *Learning Dynamical Systems from Partial Observations*, preprint, arXiv:1902.11136, 2019.

[7] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2006.

[8] M. Bocquet, J. Brajard, A. Carrassi, and L. Bertino, *Data assimilation as a learning tool to infer ordinary differential equation representations of dynamical models*, Nonlinear Processes in Geophysics, 26 (2019), pp. 143–162.

[9] M. Bocquet, J. Brajard, A. Carrassi, and L. Bertino, *Bayesian Inference of Chaotic Dynamics by Merging Data Assimilation, Machine Learning and Expectation-Maximization*, preprint, arXiv:2001.06270, 2020.

[10] M. Bocquet, C. A. Pires, and L. Wu, *Beyond Gaussian statistical modeling in geophysical data assimilation*, Monthly Weather Review, 138 (2010), pp. 2997–3023.
[11] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, *JAX: Composable Transformations of Python+NumPy Programs*, http://github.com/google/jax, 2018.
[12] J. Brajard, A. Carrassi, M. Bocquet, and L. 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*, J. Comput. Sci., 44 (2020), 101171.
[13] S. L. Brunton and J. N. Kutz, *Data-driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, Cambridge University Press, Cambridge, UK, 2019.
[14] A. Carrassi, M. Bocquet, A. Hannart, and M. Ghil, *Estimating model evidence using data assimilation*, Quart. J. Royal Meteorological Society, 143 (2017), pp. 866–880.
[15] N. K. Chada, Y. Chen, and D. Sanz-Alonso, *Iterative ensemble kalman methods: A unified perspective with some new variants*, Foundations of Data Science, 3 (2021), pp. 331–369.
[16] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, *Neural ordinary differential equations*, in Advances in Neural Information Processing Systems, Vol. 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds., Curran Associates, 2018, pp. 6571–6583, https://proceedings.neurips.cc/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf.
[17] T. J. Cocucci, M. Pulido, M. Lucini, and P. Tandeo, *Model error covariance estimation in particle and ensemble Kalman filters using an online expectation–maximization algorithm*, Quart. J. Royal Meteorological Society, 147 (2021), pp. 526–543.
[18] A. Corenflos, J. Thornton, A. Doucet, and G. Deligiannidis, *Differentiable Particle Filtering via Entropy-Regularized Optimal Transport*, preprint, arXiv:2102.07850, 2021.
[19] D. Crisan, P. Del Moral, A. Jasra, and H. Ruzayqat, *Log-Normalization Constant Estimation Using the Ensemble Kalman-Bucy Filter with Application to High-Dimensional Models*, preprint, arXiv:2101.11460, 2021.
[20] E. De Brouwer, J. Simm, A. Arany, and Y. Moreau, *GRU-ODE-Bayes: Continuous Modeling of Sporadically-Observed Time Series*, preprint, arXiv:1905.12374, 2019.
[21] P. Del Moral, *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*, Probab. Appl., Springer-Verlag, New York, 2004.
[22] P. Del Moral, J. Tugaut, et al., *On the stability and the uniform propagation of chaos properties of ensemble Kalman–Bucy filters*, Ann. Appl. Probab., 28 (2018), pp. 790–850.
[23] T. DelSole and X. Yang, *State and parameter estimation in stochastic dynamical models*, Phys. D, 239 (2010), pp. 1781–1788.
[24] A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, J. R. Stat. Soc. Ser. B Stat. Methodol., 39 (1977), pp. 1–22.
[25] A. Doucet, S. Godsill, and C. Andrieu, *On sequential Monte Carlo sampling methods for Bayesian filtering*, Statist. Comput., 10 (2000), pp. 197–208.
[26] A. Doucet and A. M. Johansen, *A tutorial on particle filtering and smoothing: Fifteen years later*, Handbook Nonlinear Filtering, 12 (2009), 3.
[27] D. Dreano, P. Tandeo, M. Pulido, B. Ait-El-Fquih, T. Chonavel, and I. Hoteit, *Estimating model-error covariances in nonlinear state-space models using Kalman smoothing and the expectation–maximization algorithm*, Quart. J. Royal Meteorological Society, 143 (2017), pp. 1877–1885.
[28] C. Drovandi, R. G. Everitt, A. Golightly, D. Prangle, et al., *Ensemble MCMC: Accelerating pseudo-marginal MCMC for state space models using the ensemble Kalman filter*, Bayesian Anal., to appear.
[29] G. Evensen, *Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics*, J. Geophysical Research Oceans, 99 (1994), pp. 10143–10162.
[30] G. Evensen, *Data Assimilation: The Ensemble Kalman Filter*, Springer, New York, 2009.
[31] M. Fraccaro, S. Kamronn, U. Paquet, and O. Winther, *A Disentangled Recognition and Nonlinear Dynamics Model for Unsupervised Learning*, preprint, arXiv:1710.05741, 2017.
[32] R. Furrer and T. Bengtsson, *Estimation of high-dimensional prior and posterior covariance matrices in Kalman filter variants*, J. Multivariate Anal., 98 (2007), pp. 227–255.

[33] G. Gaspari and S. E. Cohn, *Construction of correlation functions in two and three dimensions*, Quart. J. Royal Meteorological Society, 125 (1999), pp. 723–757.
[34] M. B. Giles, *Collected matrix derivative results for forward and reverse mode algorithmic differentiation*, in Advances in Automatic Differentiation, Springer, New York, 2008, pp. 35–44.
[35] N. J. Gordon, D. J. Salmond, and A. F. Smith, *Novel approach to nonlinear/non-Gaussian Bayesian state estimation*, IEE Proc. F Radar and Signal Processing, 140 (1993), pp. 107–113.
[36] G. A. Gottwald and S. Reich, *Supervised learning from noisy observations: Combining machine-learning techniques with data assimilation*, Phys. D, 423 (2021), 132911.
[37] T. M. Hamill, J. S. Whitaker, and C. Snyder, *Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter*, Monthly Weather Review, 129 (2001), pp. 2776–2790.
[38] A. Hannart, A. Carrassi, M. Bocquet, M. Ghil, P. Naveau, M. Pulido, J. Ruiz, and P. Tandeo, *DADA: Data assimilation for the detection and attribution of weather and climate-related events*, Climatic Change, 136 (2016), pp. 155–174.
[39] J. Harlim, S. W. Jiang, S. Liang, and H. Yang, *Machine learning for prediction with missing dynamics*, J. Comput. Phys., 428 (2021), 109922.
[40] P. L. Houtekamer and H. L. Mitchell, *Data assimilation using an ensemble Kalman filter technique*, Monthly Weather Review, 126 (1998), pp. 796–811.
[41] P. L. Houtekamer and H. L. Mitchell, *A sequential ensemble Kalman filter for atmospheric data assimilation*, Monthly Weather Review, 129 (2001), pp. 123–137.
[42] P. L. Houtekamer and F. Zhang, *Review of the ensemble Kalman filter for atmospheric data assimilation*, Monthly Weather Review, 144 (2016), pp. 4489–4532.
[43] T. Ishizone, T. Higuchi, and K. Nakamura, *Ensemble Kalman Variational Objectives: Nonlinear Latent Trajectory Inference with a Hybrid of Variational Inference and Ensemble Kalman Filter*, preprint, arXiv:2010.08729, 2020.
[44] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, Courier Corporation, 2007.
[45] R. E. Kalman, *A new approach to linear filtering and prediction problems*, Trans. ASME J. Basic Engineering, 82 (1960).
[46] M. Katzfuss, J. R. Stroud, and C. K. Wikle, *Understanding the ensemble Kalman filter*, Amer. Statist., 70 (2016), pp. 350–357.
[47] M. Katzfuss, J. R. Stroud, and C. K. Wikle, *Ensemble Kalman methods for high-dimensional hierarchical dynamic space-time models*, J. Amer. Statist. Assoc., 115 (2020), pp. 866–885.
[48] D. P. Kingma and M. Welling, *Auto-Encoding Variational Bayes*, in Proceedings of the 2nd International Conference on Learning Representations, 2014.
[49] A. Kloss, G. Martius, and J. Bohg, *How to train your differentiable filter*, Autonomous Robots, (2021), pp. 1–18.
[50] R. Krishnan, U. Shalit, and D. Sontag, *Structured inference networks for nonlinear state space models*, in Proceedings of the AAAI Conference on Artificial Intelligence, 2017.
[51] E. Kwiatkowski and J. Mandel, *Convergence of the square root ensemble Kalman filter in the large ensemble limit*, SIAM/ASA J. Uncertain. Quantif., 3 (2015), pp. 1–17.
[52] K. J. Law, D. Sanz-Alonso, A. Shukla, and A. M. Stuart, *Filter accuracy for the Lorenz 96 model: Fixed versus adaptive observation operators*, Phys. D, 325 (2016), pp. 1–13.
[53] K. J. Law and A. M. Stuart, *Evaluating data assimilation algorithms*, Monthly Weather Review, 140 (2012), pp. 3757–3782.
[54] K. J. Law, A. M. Stuart, and K. Zygalakis, *Data Assimilation*, Springer, New York, 2015.
[55] K. J. Law, H. Tembine, and R. Tempone, *Deterministic mean-field ensemble Kalman filtering*, SIAM J. Sci. Comput., 38 (2016), pp. A1251–A1279.
[56] T. A. Le, M. Igl, T. Rainforth, T. Jin, and F. Wood, *Auto-Encoding sequential Monte Carlo*, in Proceedings of the International Conference on Learning Representations, 2018, https://openreview.net/forum?id=BJ8c3f-0b.
[57] F. Le Gland and L. Mevel, *Recursive identification in hidden markov models*, in Proceedings of the 36th Conference on Decision and Control, Vol. 4, 1997, pp. 3468–3473.
[58] F. Le Gland, V. Monbet, and V.-D. Tran, *Large Sample Asymptotics for the Ensemble Kalman Filter*, Ph.D. thesis, INRIA, 2009.

[59] M. E. Levine and A. M. Stuart, *A Framework for Machine Learning of Model Error in Dynamical Systems*, preprint, arXiv:2107.06658, 2021.
[60] E. N. Lorenz, *Predictability: A problem partly solved*, in Proceedings of the Seminar on Predictability, 1996.
[61] C. J. Maddison, D. Lawson, G. Tucker, N. Heess, M. Norouzi, A. Mnih, A. Doucet, and Y. W. Teh, *Filtering Variational Objectives*, preprint, arXiv:1705.09279, 2017.
[62] A. J. Majda and J. Harlim, *Filtering Complex Turbulent Systems*, Cambridge University Press, Cambridge, UK, 2012.
[63] H. P. McKean, *Propagation of chaos for a class of non-linear parabolic equations*, in Stochastic Differential Equations, Lecture Series in Differential Equations, Session 7, 1967, pp. 41–57.
[64] S. Metref, A. Hannart, J. Ruiz, M. Bocquet, A. Carrassi, and M. Ghil, *Estimating model evidence using ensemble-based data assimilation with localization—The model selection problem*, Quart. J. Royal Meteorological Society, 145 (2019), pp. 1571–1588.
[65] C. Naesseth, S. Linderman, R. Ranganath, and D. Blei, *Variational sequential Monte Carlo*, in International Conference on Artificial Intelligence and Statistics, PMLR, 2018, pp. 968–977.
[66] D. Nguyen, S. Ouala, L. Drumetz, and R. Fablet, *EM-Like Learning Chaotic Dynamics from Noisy and Partial Observations*, preprint, arXiv:1903.10335, 2019.
[67] E. Ott, B. R. Hunt, I. Szunyogh, A. V. Zimin, E. J. Kostelich, M. Corazza, E. Kalnay, D. Patil, and J. A. Yorke, *A local ensemble Kalman filter for atmospheric data assimilation*, Tellus A Dynamic Meteorology and Oceanography, 56 (2004), pp. 415–428.
[68] O. Papaspiliopoulos and M. Ruggiero, *Optimal filtering and the dual process*, Bernoulli, 20 (2014), pp. 1999–2019.
[69] R. Pascanu, T. Mikolov, and Y. Bengio, *On the difficulty of training recurrent neural networks*, in International Conference on Machine Learning, PMLR, 2013, pp. 1310–1318.
[70] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., *PyTorch: An imperative style, high-performance deep learning library*, in Advances in Neural Information Processing Systems, 32, 2019.
[71] M. Pulido, P. Tandeo, M. Bocquet, A. Carrassi, and M. Lucini, *Stochastic parameterization identification using ensemble Kalman filtering combined with maximum likelihood methods*, Tellus A Dynamic Meteorology and Oceanography, 70 (2018), pp. 1–17.
[72] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Multistep Neural Networks for Data-Driven Discovery of Nonlinear Dynamical Systems*, preprint, arXiv:1801.01236, 2018.
[73] R. Ranganath, S. Gerrish, and D. Blei, *Black box variational inference*, in Artificial Intelligence and Statistics, PMLR, 2014, pp. 814–822.
[74] S. S. Rangapuram, M. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski, *Deep state space models for time series forecasting*, in Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 7796–7805.
[75] S. Reich and C. Cotter, *Probabilistic Forecasting and Bayesian Data Assimilation*, Cambridge University Press, Cambridge, UK, 2015.
[76] D. J. Rezende, S. Mohamed, and D. Wierstra, *Stochastic backpropagation and approximate inference in deep generative models*, in International Conference on Machine Learning, PMLR, 2014, pp. 1278–1286.
[77] M. Roth, G. Hendeby, C. Fritsche, and F. Gustafsson, *The Ensemble Kalman filter: A signal processing perspective*, EURASIP J. Advances in Signal Processing, 2017 (2017), pp. 1–16.
[78] Y. Rubanova, R. T. Chen, and D. K. Duvenaud, *Latent ordinary differential equations for irregularly-sampled time series*, Advances in Neural Information Processing Systems, 32, 2019.
[79] P. Sakov and L. Bertino, *Relation between two common localisation methods for the EnKF*, Comput. Geosci., 15 (2011), pp. 225–237.
[80] D. Sanz-Alonso, *Importance sampling and necessary sample size: An information theory approach*, SIAM/ASA J. Uncertain. Quantif., 6 (2018), pp. 867–879.
[81] D. Sanz-Alonso and A. M. Stuart, *Long-time asymptotics of the filtering distribution for partially observed chaotic dynamical systems*, SIAM/ASA J. Uncertain. Quantif., 3 (2015), pp. 1200–1220.
[82] D. Sanz-Alonso, A. M. Stuart, and A. Taeb, *Inverse Problems and Data Assimilation*, preprint, arXiv:1810.06191, 2019.

[83] D. Sanz-Alonso and Z. Wang, *Bayesian update with importance sampling: Required sample size*, Entropy, 23 (2021), https://doi.org/10.3390/e23010022.
[84] C. Schillings and A. M. Stuart, *Analysis of the ensemble Kalman filter for inverse problems*, SIAM J. Numer. Anal., 55 (2017), pp. 1264–1290.
[85] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson, *Obstacles to high-dimensional particle filtering*, Monthly Weather Review, 136 (2008), pp. 4629–4640.
[86] J. R. Stroud and T. Bengtsson, *Sequential state and variance estimation within the ensemble Kalman filter*, Monthly Weather Review, 135 (2007), pp. 3194–3208.
[87] J. R. Stroud, M. Katzfuss, and C. K. Wikle, *A Bayesian adaptive ensemble Kalman filter for sequential state and parameter estimation*, Monthly Weather Review, 146 (2018), pp. 373–386.
[88] J. R. Stroud, M. L. Stein, B. M. Lesht, D. J. Schwab, and D. Beletsky, *An ensemble Kalman filter and smoother for satellite data assimilation*, J. Amer. Statist. Assoc., 105 (2010), pp. 978–990.
[89] I. Sutskever, O. Vinyals, and Q. V. Le, *Sequence to sequence learning with neural networks*, in Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.
[90] A.-S. Sznitman, *Topics in propagation of chaos*, in Ecole d'été de probabilités de Saint-Flour XIX—1989, Springer, New York, 1991, pp. 165–251.
[91] I. Szunyogh, E. J. Kostelich, G. Gyarmati, E. Kalnay, B. R. Hunt, E. Ott, E. Satterfield, and J. A. Yorke, *A local ensemble transform Kalman filter data assimilation system for the NCEP global model*, Tellus A Dynamic Meteorology and Oceanography, 60 (2008), pp. 113–130.
[92] P. Tandeo, M. Pulido, and F. Lott, *Offline parameter estimation using EnKF and maximum likelihood error covariance estimates: Application to a subgrid-scale orography parametrization*, Quart. J. Royal Meteorological Society, 141 (2015), pp. 383–395.
[93] G. Ueno and N. Nakamura, *Iterative algorithm for maximum-likelihood estimation of the observation-error covariance matrix for ensemble-based filters*, Quart. J. Royal Meteorological Society, 140 (2014), pp. 295–315.
[94] G. Ueno and N. Nakamura, *Bayesian estimation of the observation-error covariance matrix in ensemble-based filters*, Quart. J. Royal Meteorological Society, 142 (2016), pp. 2055–2080.
[95] G. C. Wei and M. A. Tanner, *A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms*, J. Amer. Statist. Assoc., 85 (1990), pp. 699–704.
[96] J. S. Whitaker, T. M. Hamill, X. Wei, Y. Song, and Z. Toth, *Ensemble data assimilation with the ncep global forecast system*, Monthly Weather Review, 136 (2008), pp. 463–482.
[97] R. J. Williams and D. Zipser, *Gradient-based learning algorithms for recurrent networks and their computational complexity*, in Back-propagation: Theory, Architectures and Applications, Y. Chauvin and D. E. Rumelhart, eds., Erlbaum, Hillsdale, NJ, 1995, pp. 433–486.
[98] K. Xu and C. K. Wikle, *Estimation of parameterized spatio-temporal dynamic models*, J. Statist. Plann. Inference, 137 (2007), pp. 567–588.