Affine-mapping based variational ensemble Kalman filter

Wen, Linjie; Li, Jinglai

DOI:
10.1007/s11222-022-10156-5

Citation for published version (Harvard):
Wen, L & Li, J 2022, 'Affine-mapping based variational ensemble Kalman filter', *Statistics and Computing*, vol. 32, no. 6, 97. https://doi.org/10.1007/s11222-022-10156-5

Statistics and Computing (2022) 32:97
https://doi.org/10.1007/s11222-022-10156-5

ORIGINAL PAPER

Affine-mapping based variational ensemble Kalman filter

Linjie Wen[1] · Jinglai Li[2]

Received: 6 June 2022 / Accepted: 18 September 2022

Abstract

We propose an affine-mapping based variational ensemble Kalman filter for sequential Bayesian filtering problems with generic observation models. Specifically, the proposed method is formulated as to construct an affine mapping from the prior ensemble to the posterior one, and the affine mapping is computed via a variational Bayesian formulation, i.e., by minimizing the Kullback–Leibler divergence between the transformed distribution through the affine mapping and the actual posterior. Some theoretical properties of resulting optimization problem are studied and a gradient descent scheme is proposed to solve the resulting optimization problem. With numerical examples we demonstrate that the method has competitive performance against existing methods.

Keywords Affine-mapping · Data assimilation · Ensemble Kalman Filters · Kullback–Leibler divergence · Sequential Bayesian filtering

Mathematics Subject Classification 65C05 · 62F15

1 Introduction

The ensemble Kalman filter (EnKF) (Evensen 2009, 2003) is one of the most popular tools for sequential data assimilation, thanks to its computational efficiency and flexibility (Houtekamer and Mitchell 1998; Whitaker and Hamill 2002; Evensen 2003). Simply put, at each time step EnKF approximates the prior, the likelihood and the posterior by Gaussian distributions. Such a Gaussian approximation allows an affine update that maps the prior ensemble to the posterior one. This Gaussian approximation and the resulting affine update are the key that enables EnKF to handle large-scale problems with a relatively small number of ensembles. In the conventional EnKF, it is required that the observation model is Gaussian-linear, which means that the observation operator is linear and the noise is additive Gaussian. However, in many real-world applications, neither of these two requirements is satisfied. When the actual observation model is not Gaussian-linear, the EnKF method may suffer from substantial estimation error, which is discussed in details in Sect. 3.2.

We note that, many EnKF variants (see, e.g., Law et al. 2015 and the references therein), such as the ensemble transform Kalman filter (ETKF) (Bishop et al. 2001), are mainly designed to improve the performance of EnKF under the standard Gaussian-linear observation model, and thus have the same difficulty with non-Gaussian-linear observation models. To the end, it is of practical importance to develop methods that can better deal with generic observation models than EnKF, while retaining the computational advantage (i.e., using a small ensemble size) of it.

A notable example of such methods is the nonlinear ensemble adjustment filter (NLEAF) (Lei and Bickel 2011), which involves a correction scheme: the posterior moments are calculated with importance sampling and the ensembles are then corrected accordingly. Another very interesting class of methods are the (conditional) mean-field EnKF (Law et al. 2016; Hoang et al. 2021), which is derived via the formulation of computing an optimal point estimator in the mean-square error sense. The mean-field methods can outperform the standard EnKF in many applications, but they still require certain assumptions on the observation noise. Other methods that can be applied to such problems include (Anderson 2003,

✉ Jinglai Li
j.li.10@bham.ac.uk

Linjie Wen
wenlinjie@pku.edu.cn

[1] School of Earth and Space Sciences, Peking University, 5 Yiheyuan Rd, Beijing 100871, China

[2] School of Mathematics, University of Birmingham, Birmingham B15 2TT, UK

Published online: 19 October 2022

2001; Houtekamer and Mitchell 2001; Li et al. 2018; Ba et al. 2018) (some of them may need certain modifications), just to name a few. In this work we focus on the EnKF type of methods that can use a small number of ensembles in high dimensional problems, and methods involving full Monte Carlo sampling such as the particle filter (PF) (Arulampalam et al. 2002; Doucet and Johansen 2009), or those seeking to compute the exact posterior through transport maps (Spantini et al. 2019), are not in our scope. It is also worth noting that a class of methods combine EnKF and PF to alleviate the estimation bias induced by the non-Gaussianity (e.g., Stordal et al. 2011; Frei and Künsch 2013), and typically the EnKF part in such methods still requires a Gaussian-linear observation model (or to be treated as such a model).

The main purpose of this work is to provide an alternative framework to implement EnKF for arbitrary observation models. Specifically, the proposed method formulates the EnKF update as to construct an affine mapping from the prior to the posterior and such an affine mapping is computed in variational Bayesian framework (MacKay 2003). That is, we seek the affine mapping minimizing the Kullback–Leibler divergence (KLD) between the "transformed" prior distribution and the posterior. We note here that a similar formulation has been used in the variational (ensemble) Kalman filter (Auvinen et al. 2010; Solonen et al. 2012). The difference is however, the variational (ensemble) Kalman filter methods mentioned above still rely on the linear-Gaussian observation model, where the variational formulation, combined with a BFGS scheme, is used to avoid the inversion and storage of very large matrices, while in our work the variational formulation is used to compute the optimal affine mapping for generic observation models.

It can be seen that this affine mapping based variational EnKF (VEnKF) reduces to the standard EnKF when the observation model is Gaussian-linear, and as such it is a natural generalization of the standard EnKF to generic observation models. Also, by design the obtained affine mapping is *optimal* under the variational (minimal KLD) principle. We also present a numerical scheme based on gradient descent algorithm to solve the resulting optimization problem, and with numerical examples we demonstrate that the method has competitive performance against several existing methods. Finally we emphasize that, as an extension of EnKF, the proposed method also requires that the prior and the posterior distributions should not deviate significantly from Gaussian.

The rest of the work is organized as follows. In Sect. 2 we provide a generic formulation of the sequential Bayesian filtering problem. In Sect. 3 we present the proposed affine mapping based variational EnKF. Numerical examples are provided in Sect. 4 to demonstrate the performance of the proposed method and finally some closing remarks are offered in Sect. 5.

2 Problem formulation

2.1 Hidden Markov model

We start with the hidden Markov model (HMM), which is a generic formulation for data assimilation problems (Doucet and Johansen 2009). Specifically let $\{x_t\}_{t\geq 0}$ and $\{y_t\}_{t\geq 0}$ be two discrete-time stochastic processes, taking values from continuous state spaces $\mathcal{X}$ and $\mathcal{Y}$ respectively. Throughout this work we assume that $\mathcal{X} = \mathbb{R}^{n_x}$ and $\mathcal{Y} = \mathbb{R}^{n_y}$. The HMM model assumes that the pair $\{x_t, y_t\}$ has the following property,

$$x_t|x_{1:t-1}, y_{1:t-1} \sim \pi(x_t|x_{t-1}), \quad x_0 \sim \pi(x_0), \tag{1a}$$

$$y_t|x_{1:t}, y_{1:t-1} \sim \pi(y_t|x_t), \tag{1b}$$

where for simplicity we assume that the probability density functions (PDF) of all the distributions exist and $\pi(\cdot)$ is used as a generic notation of a PDF whose actual meaning is specified by its arguments.

In the HMM formulation, $\{x_t\}_{t\geq 0}$ and $\{y_t\}_{t\geq 0}$ are known respectively as the hidden and the observed states, and a schematic illustration of HMM is shown in Fig. 1. This framework represents many practical problems of interest (Fine et al. 1998; Krogh et al. 2001; Beal et al. 2002), where one makes observations of $\{y_t\}_{t\geq 0}$ and wants to estimate the hidden states $\{x_t\}_{t\geq 0}$ therefrom. A typically example of HMM is the following stochastic discrete-time dynamical system:

$$x_t = F_t(x_{t-1}, \alpha_t), \quad x_0 \sim \pi(x_0), \tag{2a}$$

$$y_t = G_t(x_t, \beta_t), \tag{2b}$$

where $\alpha_t \sim \pi_t^{\alpha}(\cdot)$ and $\beta_t \sim \pi_t^{\beta}(\cdot)$ are random variables representing respectively the model error and the observation noise at time t. In many real-world applications such as numerical weather prediction (Bauer et al. 2015), Eq. (2a), which represents the underlying physical model, is computationally intensive, while Eq. (2b), describing the observation model, is available analytically and therefore easy to evaluate. It follows that, in such problems, (1) one can only afford a small number of particles in the filtering, (2) Eq. (2a) accounts for the vast majority of the computational cost.

All our numerical examples are described in this form and further details can be found in Sect. 4.

2.2 Recursive Bayesian filtering

Recursive Bayesian filtering (Chen 2003) is a popular framework to estimate the hidden states in a HMM, and it aims to compute the condition distribution $\pi(x_t|y_{1:t})$ for $t = 1, 2, \ldots$ recursively. In what follows we discuss how the recursive Bayesian filtering proceeds.

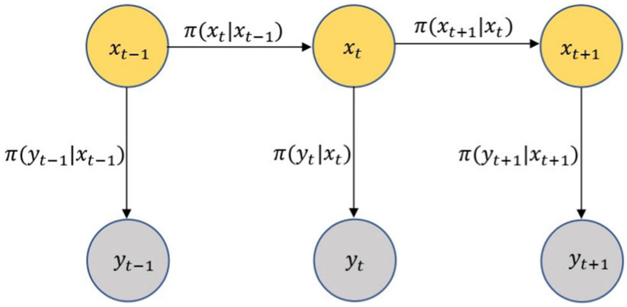


Fig. 1 A schematic illustration of the Hidden Markov Model

First applying the Bayes' formula, we obtain

$$\pi(x_t|y_{1:t}) = \frac{\pi(y_t|x_t, y_{1:t-1})\pi(x_t|y_{1:t-1})}{\pi(y_t|y_{1:t-1})}, \tag{3}$$

where $\pi(y_t|y_{1:t-1})$ is the normalization constant (Doucet and Johansen 2009). From Eq. (1b) we know that y_t is independent of y_{t-1} conditionally on x_t, and thus Eq. (3) becomes

$$\pi(x_t|y_{1:t}) = \frac{\pi(y_t|x_t)\pi(x_t|y_{1:t-1})}{\pi(y_t|y_{1:t-1})}. \tag{4}$$

The condition distribution $\pi(x_t|y_{1:t-1})$ can be expressed as

$$\pi(x_t|y_{1:t-1}) = \int \pi(x_t|x_{t-1}, y_{1:t-1})\pi(x_{t-1}|y_{1:t-1})dx_{t-1}, \tag{5}$$

and again thanks to the property of the HMM in Eq. (1), we have (Doucet and Johansen 2009),

$$\pi(x_t|y_{1:t-1}) = \int \pi(x_t|x_{t-1})\pi(x_{t-1}|y_{1:t-1})dx_{t-1}, \tag{6}$$

where $\pi(x_{t-1}|y_{1:t-1})$ is the posterior distribution at the previous step $t-1$.

As a result the recursive Bayesian filtering performs the following two steps in each iteration:

- Prediction step: the prior density $\pi(x_t|y_{1:t-1})$ is determined via Eq. (6),
- Update step: the posterior density $\pi(x_t|y_{1:t})$ is computed via Eq. (4).

The recursive Bayesian filtering provides a generic framework for sequentially computing the conditional distribution $\pi(x_t|y_{1:t})$ as the iteration proceeds. In practice, the analytical expressions for the posterior $\pi(x_t|y_{1:t})$ or the prior $\pi(x_t|y_{1:t-1})$ usually can not be obtained, and therefore these distributions have to be represented numerically, for example, by an ensemble of particles (Doucet and Johansen 2009).

3 Affine-mapping based VEnKF

We describe the affine-mapping based VEnKF (AM-VEnKF) algorithm in this section.

3.1 Formulation of the affine-mapping based VEnKF

We first consider the update step: namely suppose that the prior distribution $\pi(x_t|y_{1:t-1})$ is obtained, and we want to compute the posterior $\pi(x_t|y_{1:t})$.

We start with a brief introduction to the transport map based methods for computing the posterior distribution (El Moselhy and Marzouk 2012), where the main idea is to construct a mapping which pushes the prior distribution into the posterior. Namely suppose $\tilde{x}_t$ follows the prior distribution $\pi(\cdot|y_{1:t-1})$, and one aims to construct a bijective mapping $T : \mathcal{X} \to \mathcal{X}$,

such that $x_t = T(\tilde{x}_t)$ follows the posterior distribution $\pi(\cdot|y_{1:t})$. In reality, it is usually infeasible to obtain the mapping that can exactly push the prior into the posterior $\pi(\cdot|y_{1:t})$, and in this case an approximate approach can be used. That is, let $\pi_T(\cdot)$ be the distribution of $x_t = T(\tilde{x}_t)$ where $\tilde{x}_t \sim \pi(\cdot|y_{1:t-1})$ and we seek a mapping $T \in \mathcal{H}$ where $\mathcal{H}$ is a given function space, so that $\pi_T(\cdot)$ is "closest" to the actual posterior $\pi(\cdot|y_{1:t})$ in terms of certain measure of distance between two distributions.

In practice, the KLD, which (for any two distributions π_1 and π_2) is defined as,

$$\mathcal{D}_{\text{KL}}(\pi_1, \pi_2) = \int \log\left[\frac{\pi_1(x)}{\pi_2(x)}\right]\pi_1(x)dx, \tag{7}$$

is often used for such a distance measure. That is, we find a mapping T by solving the following minimization problem,

$$\min_{T\in\mathcal{H}} \mathcal{D}_{\text{KL}}(\pi_T, \pi(x_t|y_{1:t})), \tag{8}$$

which can be understood as a variational Bayes formulation (Wainwright and Jordan 2008).

In practice, the prior distribution $\pi(\tilde{x}_t|y_{1:t-1})$ is usually not analytically available, and in particular they are represented by an ensemble of particles. As is in the standard EnKF, we estimate a Gaussian approximation of the prior distribution $\pi(\tilde{x}_t|y_{1:t-1})$ from the ensemble. Namely, given an ensemble $\{\tilde{x}_t^m\}_{m=1}^M$ drawn from the prior distribution $\hat{\pi}(\tilde{x}_t|y_{1:t-1})$, we construct an approximate prior $\hat{\pi}(\cdot|y_{1:t-1}) = N(\tilde{\mu}_t, \tilde{\Sigma}_t)$, with

$$\tilde{\mu}_t = \frac{1}{M}\sum_{m=1}^{M} \tilde{x}_t^m,$$

$$\tilde{\Sigma}_t = \frac{1}{M-1}\sum_{m=1}^{M} (\tilde{x}_t^m - \tilde{\mu}_t)(\tilde{x}_t^m - \tilde{\mu}_t)^T. \tag{9}$$

As a result, Eq. (8) is modified to

$$\min_{T\in\mathcal{H}} \mathcal{D}_{\mathrm{KL}}(\pi_T, \hat{\pi}(x_t|y_{1:t})), \quad \text{with}$$

$$\hat{\pi}(\cdot|y_{1:t}) \propto \hat{\pi}(\cdot|y_{1:t-1})\pi(y_t|x_t). \tag{10}$$

Namely, we seek to minimize the distance between π_T and the approximate posterior $\hat{\pi}(x_t|y_{1:t})$. We refer to the filtering algorithm by solving Eq. (10) as VEnKF, where the complete algorithm is given in Alg. 1.

Algorithm 1 Affine-mapping based variational ensemble Kalman filter (AM-VEnKF)

```
– Prediction:
    – Let $\tilde{x}_t^m \sim f_t(\cdot|x_{t-1}^m)$, $m = 1, 2, \ldots, M$;
    – Let $\hat{\pi}(\cdot|y_{1:t-1}) = N(\tilde{\mu}_t, \hat{\Sigma}_t)$ where $\tilde{\mu}_t$ and $\hat{\Sigma}_t$ are computed
      using Eq. (9);
– Update:
    – Let $\hat{\pi}(x_t|y_{1:t}) \propto \hat{\pi}(x_t|y_{1:t-1})\pi(y_t|x_t)$;
    – Solve the minimization problem:
          $T_t = \arg\min_{T\in\mathcal{H}} \mathcal{D}_{\mathrm{KL}}(\pi_T, \hat{\pi}(x_t|y_{1:t}))$.
    – Let $x_t^m = T_t\tilde{x}_t^m$ for $m = 1, \ldots, M$.
```

Now a key issue is to specify a suitable function space $\mathcal{H}$. First let A and b be $n_x \times n_x$ and $n_x \times 1$ matrices respectively, and we can define a space of affine mappings $\mathcal{A} = \{T : T\cdot = A\cdot + b\}$, with norm $\|T\| = \sqrt{\|A\|_2^2 + \|b\|_2^2}$. Now we choose

$$\mathcal{H} = \{T \in \mathcal{A} \,|\, \|T\| \le r, \ \mathrm{rank}(A) = n_x\},$$

where r is any fixed positive constant. It is obvious that A being full-rank implies that T is invertible, which is an essential requirement for the proposed method, and will be discussed in detail in Sect. 3.3.

Next we show that the minimizer of KLD exists in the closure of $\mathcal{H}$:

Theorem 1 *Let P and Q be two arbitrary probability distributions defined on a Borel set $\mathcal{B}(\mathbb{R}^{n_x})$, and*

$$\mathcal{H}^* = \{T \in \mathcal{A} \,|\, \|T\| \le r\},$$

for some fixed $r > 0$. Let P_T be the distribution of $T(x)$, given x being a $\mathbb{R}^{n_x}$-valued random variable following P. The functional $\mathcal{D}_{\mathrm{KL}}(P_T, Q)$ on $\mathcal{H}^*$ admits a minimizer.

Proof Let $\Omega = \{P_T : T \in \mathcal{H}^*\}$ be the image of $\mathcal{H}^*$ into $\mathcal{P}(\mathbb{R}^{n_x})$, the space of all Borel probability measures on $\mathbb{R}^{n_x}$. For any sequence $\{T_n\}_{n=1}^{\infty} \in \mathcal{H}^*$ and $T \in \mathcal{H}^*$ such that $T_n \to T$, we have that $T_n(x) \to T(x)$ almost surely (a.s.), which implies that P_{T_n} converges to P_T weakly. It follows directly that P_T is continuous on $\mathcal{H}^*$.

Since $\mathcal{H}^*$ is a compact subset of $\mathcal{A}$, its image Ω is compact in $\mathcal{P}(\mathbb{R}^{n_x})$. Since $\mathcal{D}_{\mathrm{KL}}(P_T, Q)$ is lower semi-continuous with respect to P_T [Theorem 1 in Posner (1975)], $\min_{P_T\in\Omega} \mathcal{D}_{\mathrm{KL}}(P_T, Q)$ admits a solution P_{T^*} with $T^* \in \mathcal{H}^*$. It follows that T^* is a minimizer of $\min_{T\in\mathcal{H}^*} \mathcal{D}_{\mathrm{KL}}(P_T, Q)$. □

Finally it is also worth mentioning that, a key assumption of the proposed method (and EnKF as well) is that both the prior and posterior ensembles should not deviate strongly from Gaussian. To this end, a natural requirement for the chosen function space $\mathcal{H}$ is that, for any $T \in \mathcal{H}$, if $\pi(\tilde{x}_t|y_{1:t-1})$ is close to Gaussian, so should be $\pi_T(x_t)$ with $x_t = T(\tilde{x}_t)$. Obviously an arbitrarily function space does not satisfy such a requirement. However, for affine mappings, we have the following proposition:

Proposition 1 *For a given positive constant number ϵ, if there is a n_x-dimensional normal distribution $\tilde{p}_G$ such that $\mathcal{D}_{\mathrm{KL}}(\tilde{p}_G(\tilde{x}_t), \pi(\tilde{x}_t|y_{1:t-1})) < \epsilon$, and if $T \in \mathcal{H}$, there must exist a n_x-dimensional normal distribution p_G satisfying $\mathcal{D}_{\mathrm{KL}}(p_G(x_t), \pi_T(x_t)) < \epsilon$.*

Proof This proposition is a direct consequence of the fact that KLD is invariant under affine transformations. □

Loosely the proposition states that, for an affine mapping T, if the prior $\pi(\tilde{x}_t|y_{1:t-1})$ is close to a Gaussian distribution, so is $\pi_T(x_t)$, which ensures that the update step will not increase the "non-Gaussianity" of the ensemble.

In principle one can choose a different function space $\mathcal{H}$, and for example, a popular transport-based approach called the Stein variational gradient descent (SVGD) method (Liu and Wang 2016) constructs such a function space using the reproducing kernel Hilbert space (RKHS), which can also be used in the VEnKF formulation. We provide a detailed description of the SVGD based VEnKF in "Appendix A", and this method is also compared with the proposed AM-VEnKF in all the numerical examples.

3.2 Connection to the ensemble Kalman filter

In this section, we discuss the connection between the standard EnKF (Evensen 2009, 2003) and AM-VEnKF, and show that EnKF results in additional estimation error due to certain approximations made. We start with a brief introduction to EnKF. We consider the situation where the observation model takes the form of

$$y_t = \mathrm{H}_t x_t + \beta_t, \tag{11}$$

which implies $\pi(y_t|x_t) = N(H_t x_t, R_t)$, where H_t is a linear observation operator and β_t is a zero-mean Gaussian noise with covariance R_t.

In this case, EnKF can be understood as to obtain an approximate solution of Eq. (10). Recall that in the VEnKF formulation, π_T is the distribution of $x_t = T(\tilde{x}_t)$ where $\tilde{x}_t$ follows $\pi(\cdot|y_{1:t-1})$, and similarly we can define $\hat{\pi}_T$ as the distribution of $x_t = T(\tilde{x}_t)$ where $\tilde{x}_t$ follows the approximate prior $\hat{\pi}(\cdot|y_{1:t-1})$. Now instead of Eq. (10), we find T by solving,

$$\min_{T\in\mathcal{H}} \mathcal{D}_{\mathrm{KL}}(\hat{\pi}_T, \hat{\pi}(x_t|y_{1:t})), \tag{12}$$

and the obtained mapping T is then used to transform the particles. It is easy to verify that the optimal solution of Eq. (12) can be obtained exactly (Evensen 2009),

$$x_t = T(\tilde{x}_t) = (\mathrm{I} - K_t H_t)\tilde{x}_t + K_t y_t, \tag{13}$$

where I is the identity matrix and Kalman Gain matrix K_t is

$$K_t = \tilde{\Sigma}_t H_t^T (H_t \tilde{\Sigma}_t H_t^T + \mathrm{R}_t)^{-1}. \tag{14}$$

Moreover, the resulting value of KLD is zero, which means that the optimal mapping pushes the prior exactly to the posterior. One sees immediately that the optimal mapping in Eq. (13) coincides with the updating formula of EnKF, implying that EnKF is an approximation of VEnKF, even when the observation model is exactly linear-Gaussian.

When the observation model is not linear-Gaussian, further approximation is needed. Specifically the main idea is to approximate the actual observation model with a linear-Gaussian one, and estimate the Kalman gain matrix K_t directly from the ensemble (Houtekamer and Mitchell 2001). Namely, suppose we have an ensemble from the prior distribution: $\{\tilde{x}_t^m\}_{m=1}^M$, and we generate an ensemble of data points: $\tilde{y}_t^m \sim \pi(\tilde{y}_t^m|\tilde{x}_t^m)$ for $m = 1, \ldots, M$. Next we estimate the Kalman gain matrix as follows,

$$\tilde{K}_t = C_{xy} C_{yy}^{-1},$$

$$\hat{x}_t = \frac{1}{M}\sum_{m=1}^M \tilde{x}_t^m, \quad \hat{y}_t = \frac{1}{M}\sum_{m=1}^M \tilde{y}_t^m,$$

$$C_{xy} = \frac{1}{M-1}\sum_{m=1}^M (\tilde{x}_t^m - \hat{x}_t)(\tilde{y}_t^m - \hat{y}_t)^T,$$

$$C_{yy} = \frac{1}{M-1}\sum_{m=1}^M (\tilde{y}_t^m - \hat{y}_t)(\tilde{y}_t^m - \hat{y}_t)^T.$$

Finally the ensemble are updated: $x_t^m = \tilde{x}_t^m + \tilde{K}_t(y_t - \tilde{y}_t^m)$ for $i = 1, \ldots, M$. As one can see here, due to these approximations, the EnKF method can not provide an accurate solution to Eq. (10), especially when these approximations are not accurate.

3.3 Numerical algorithm for minimizing KLD

In the VEnKF framework presented in Sect. 3.1, the key step is to solve the KLD minimization problem (8). In this section we describe in details how the optimization problem is solved numerically.

Namely suppose at step t, we have a set of samples $\{\tilde{x}_t^m\}_{m=1}^M$ drawn from the prior distribution $\pi(\tilde{x}_t|y_{1:t-1})$, we want to transform them into the ensemble $\{x_t^m\}_{m=1}^M$ that follows the approximate posterior $\pi(x_t|y_{1:t})$. First we set up some notations, and for conciseness some of them are different from those used in the previous sections: first we drop the subscript of $\tilde{x}_t$ and x_t, and we then define $p(\tilde{x}) = \pi(\tilde{x}|y_{1:t-1})$ (the actual prior), $\tilde{p}(\tilde{x}) = \hat{\pi}(\tilde{x}|y_{1:t-1}) = N(\tilde{\mu}, \tilde{\Sigma})$ (the Gaussian approximate prior), $l(x) = -\log \pi(y_t|x)$ (the negative log-likelihood) and $q(x) = \hat{\pi}(x|y_{1:t})$ (the approximate posterior). It should be clear that

$$q(x) \propto \tilde{p}(x)\exp(-l(x)). \tag{15}$$

Recall that we want to minimize $\mathcal{D}_{\mathrm{KL}}(p_T(x), q(x))$ where p_T is the distribution of the transformed random variable $x = T(\tilde{x})$, and it is easy to show that

$$\mathcal{D}_{\mathrm{KL}}(p_T(x), q(x)) = \mathcal{D}_{\mathrm{KL}}(p(\tilde{x}), q_{T^{-1}}(\tilde{x})),$$

where $q_{T^{-1}}$ is the distribution of the inversely transformed random variable $\tilde{x} = T^{-1}(x)$ with $x \sim q(x)$. Moreover, as

$$\mathcal{D}_{\mathrm{KL}}(p(\tilde{x}), q_{T^{-1}}(\tilde{x})) = \int \log[p(\tilde{x})]p(\tilde{x})d\tilde{x} - \int \log[q_{T^{-1}}(\tilde{x})]p(\tilde{x})d\tilde{x},$$

minimizing $\mathcal{D}_{\mathrm{KL}}(p_T(x), q(x))$ is equivalent to

$$\min_{T\in\mathcal{H}} -\int \log[q_{T^{-1}}(\tilde{x})]p(\tilde{x})d\tilde{x}. \tag{16}$$

A difficulty here is that the feasible space $\mathcal{H}$ is constrained by $\|T\| \leq r$ (i.e. an Ivanov regularization), which poses computational challenges.

Following the convention we replace the constraint with a Tikhonov regularization to simplify the computation:

$$\min_{T\in\mathcal{A}} -\int \log[q_{T^{-1}}(\tilde{x})]p(\tilde{x})d\tilde{x} + \lambda\|T\|^2, \tag{17}$$

where λ is a pre-determined regularization constant.

Now using $Tx = Ax + b$, $q_{T^{-1}}(\tilde{x})$ can be written as,

$$q_{T^{-1}}(\tilde{x}) = q(A\tilde{x} + b)|A|, \tag{18}$$

and we substitute Eq. (18) along with Eq. (15) in to Eq. (17), yielding,

$$\begin{aligned}
&\min_{A,b} F_q(A,b)\\
&:= -\int \log[q(A\tilde{x}+b)]p(\tilde{x})d\tilde{x}\\
&\quad -\log|A| + \lambda(\|A\|_2^2 + \|b\|_2^2),\\
&= -\int \log[\tilde{p}(A\tilde{x}+b)]p(\tilde{x})d\tilde{x} + \int l(A\tilde{x}+b)p(\tilde{x})d\tilde{x}\\
&\quad -\log|A| + \lambda(\|A\|_2^2 + \|b\|_2^2),\\
&= \frac{1}{2}Tr[(\tilde{\Sigma}+\tilde{\mu}\tilde{\mu}^T)A^T\tilde{\Sigma}^{-1}A]\\
&\quad +(b-\tilde{\mu})^T\tilde{\Sigma}^{-1}[A\tilde{\mu}+\frac{1}{2}(b-\tilde{\mu})]\\
&\quad -\log|A| + \mathrm{E}_{\tilde{x}\sim p}[l(A\tilde{x}+b)]\\
&\quad +\frac{1}{2}(n_x\log(2\pi)+\log|\tilde{\Sigma}|)\\
&\quad +\lambda(\|A\|_2^2+\|b\|_2^2),
\end{aligned} \tag{19}$$

which is an unconstrained optimization problem in terms of A and b. It should be clear that the solution of Eq. (19) is naturally invertible.

We then solve the optimization problem (19) with a gradient descent (GD) scheme:

$$A_{k+1} = A_k - \epsilon_k\frac{\partial F_q}{\partial A}(A_k,b_k),$$
$$b_{k+1} = b_k - \epsilon_k\frac{\partial F_q}{\partial b}(A_k,b_k),$$

where ϵ_k is the step size and the gradients can be derived as,

$$\begin{aligned}\frac{\partial F_q}{\partial A}(A,b) &= (\tilde{\Sigma}+\tilde{\mu}\tilde{\mu}^T)A^T\tilde{\Sigma}^{-1} + \tilde{\Sigma}^{-1}(b-\tilde{\mu})\tilde{\mu}^T\\
&\quad -A^{-1} + \mathrm{E}_{\tilde{x}\sim p}[\nabla_x l(A\tilde{x}+b)\tilde{x}^T] + 2\lambda A,\end{aligned} \tag{20}$$

$$\begin{aligned}\frac{\partial F_q}{\partial b}(A,b) &= \tilde{\Sigma}^{-1}[A\tilde{\mu}+b-\tilde{\mu}]\\
&\quad +\mathrm{E}_{\tilde{x}\sim p}[\nabla_x l(A\tilde{x}+b)] + 2\lambda b.\end{aligned} \tag{21}$$

Note that Eq. (20) involves the expectations $\mathrm{E}_{\tilde{x}\sim p}[\nabla_x l(A\tilde{x}+b)\tilde{x}^T]$ and $\mathrm{E}_{\tilde{x}\sim p}[\nabla_x l(A\tilde{x}+b)]$ which are not known exactly, and in practice they can be replaced by their Monte Carlo estimates:

$$\mathrm{E}_{\tilde{x}\sim p}[\nabla_x l(A\tilde{x}+b)\tilde{x}^T] \approx \frac{1}{M}\sum \nabla_x l(A\tilde{x}^m+b)(\tilde{x}^m)^T,$$
$$\mathrm{E}_{\tilde{x}\sim p}[\nabla_x l(A\tilde{x}+b)] \approx \frac{1}{M}\sum_{m=1}^{M}\nabla_x l(A\tilde{x}^m+b),$$

where $\{\tilde{x}^m\}_{m=1}^M$ are the prior ensemble and $\nabla_x l(x)$ is the derivative of $l(x)$ taken with respect to x. The same Monte Carlo treatment also applies to the objective function $F_q(A,b)$ itself when it needs to be evaluated.

The last key ingredient of the optimization algorithm is the stopping criteria. Due to the stochastic nature of the optimization problem, standard stopping criteria in the gradient descent method are not effective here. Therefore we adopt a commonly used criterion in search-based optimization: the iteration is terminated if the current best value is not sufficiently increased within a given number of steps. More precisely, let F_k^* and $F_{k-\Delta k}^*$ be the current best value at iteration k and $k-\Delta k$ respectively where Δk is a positive integer smaller than k, and the iteration is terminated if $F_k^* - F_{k-\Delta k}^* < \Delta_F$ for a prescribed threshold Δ_F. In addition we also employ a safeguard stopping condition, which terminates the procedure after the number of iterations reaches a prescribed value $K_{\max}$.

It is also worth mentioning that the EnKF type of methods are often applied to problems where the ensemble size is similar to or even smaller than the dimensionality of the states and in this case the localization techniques are usually used to address the undersampling issue (Anderson 2007). In the AM-VEnKF method, many localization techniques developed in EnKF literature can be directly used, and in our numerical experiments we adopt the sliding-window localization used in Ott et al. (2004), and we will provide more details of this localization technique in Sect. 4.1.

Finally we provide some remarks on the theoretical property of the algorithm. First as has been mentioned, it is essential a specific implementation of the GD scheme and therefore we expect that it enjoys the same convergence property of GD from the optimization perspective. Another theoretical issue is that we here do not have results on the statistical stability of the algorithm, which is an important question and should be studied in future works.

4 Numerical examples

4.1 Observation models

In our numerical experiments, we test the proposed method with an observation model that is quite flexible and also commonly used in epidemic modeling and simulation (Capaldi et al. 2012):

$$y_t = G(x_t,\beta_t) = M(x_t) + aM(x_t)^\theta \circ \beta_t, \tag{22}$$

where $M(\cdot):\mathcal{X}\to\mathcal{Y}$ is a mapping from the state space to the observation space, a is a positive scalar, β_t is a random variable defined on $\mathcal{Y}$, and $\circ$ stands for the Schur (componentwise) product. Moreover we assume that β_t is an independent random variable with zero mean and variance R, where R here is the vector containing the variance of each component and should not be confused with the covariance matrix. It can be seen that $aM(x_t)^\theta \circ \beta_t$ represents the observation noise, controlled by two adjustable parameters θ and a, and the likelihood $\pi(y_t|x_t)$ is of mean $M(x_t)$ and variance $a^2 M(x_t)^{2\theta} \circ R$.

The parameter θ is particularly important for specifying the noise model in Capaldi et al. (2012) and here we consider the following three representative cases. First if we take $\theta = 0$, it follows that $y_t = M(x_t) + a\beta_t$, where the observation noise is independent of the state value x_t. This is the most commonly used observation model in data assimilation and we refer to it as the absolute noise following (Capaldi et al. 2012). Second if $\theta = 0.5$, the variance of observation noise is $a^2 M(x_t) \circ R$, which is linearly dependent on $M(x_t)$, and we refer to this as the Poisson noise (Capaldi et al. 2012). Finally in case of $\theta = 1$, it is the standard deviation of the noise, equal to $aM(x_t)R^{1/2}$, that depends linearly on $M(x_t)$, and this case is referred to as the relative noise (Capaldi et al. 2012). In our numerical experiments we test all the three cases.

Moreover, in the first two numerical examples provided in this work, we take

$$M(x_t) = 0.1x_t^2, \tag{23}$$

$a = 1$, and assume β_t to follow the Student's t-distribution (Roth et al. 2013) with zero-mean and variance 1.5. In the last example, we take,

$$M(x_t) = \exp(x_t/2), \tag{24}$$

and $a = 1$.

As has been mentioned, localization is needed in some numerical experiments here. Given Eqs. (23) and (24) we can see that the resulting observation model has a property that each component of the observation y_t is associated to a component of the state x_t: namely,

$$y_{t,i} = M(x_{t,i}) + (M(x_{t,i}))^\theta \beta_{t,i}, \quad i = 1, \ldots, n_x,$$

where $\beta_{t,i}$ is the i-th component of β_t, and $n_y = n_x$. In this case, we can employ the sliding-window localization method, where local observations are used to update local state vectors, and the whole state vector is reconstructed by aggregating the local updates. Namely, the state vector $x_t = (x_{t,1}, \ldots, x_{t,n_x})$ is decomposed into a number of overlapping local vectors: $\{x_{t,N_i}\}_{i=1}^{n_x}$, where $N_i = [\max\{1, i-l\} : \min\{i+l, n_x\}]$ for a positive integer l. When updating any local vector x_{t,N_i}, we only use the local observations y_{t,N_i} and as such each local vector is updated independently. It can be seen that by design each $x_{t,i}$ is updated in multiple local vectors, and the final update is calculated by averaging its updates in local vectors indexed by $N_{\max\{1,i-k\}}, \ldots, N_i, \ldots, N_{\min\{i+k,n_x\}}$, for some positive integer $k \leq l$. We refer to Ott et al. (2004), Lei and Bickel (2011) for further details.

4.2 Lorenz-96 system

Our first example is the Lorenz-96 model (Lorenz 1996):

$$\begin{cases} \frac{dx^n}{dt} = (x^{n+1} - x^{n-2})x^{n-1} - x^n + 8, \; n = 1, \ldots, 40 \\ x^0 = x^{40}, \; x^{-1} = x^{39}, \; x^{41} = x^1, \end{cases} \tag{25}$$

a commonly used benchmark example for filtering algorithms.

By integrating the system (25) via the Runge-Kutta scheme with stepsize $\Delta t = 0.05$, and adding some model noise, we obtain the following discrete-time model:

$$\begin{cases} \mathbf{x}_t = \mathcal{F}(\mathbf{x}_{t-1}) + \alpha_t, \quad t = 1, 2, \ldots \\ \mathbf{y}_t = M(\mathbf{x}_t) + M(\mathbf{x}_t)^\theta \beta_t, \quad t = 1, 2, \ldots \end{cases} \tag{26}$$

where $\mathcal{F}$ is the standard fourth-order Runge-Kutta solution of Eq. (25), α_t is standard Gaussian noise, and the initial state $\mathbf{x}_0 \sim U[0, 10]$. We use synthetic data in this example, which means that both the true states and the observed data are simulated from the model.

As mentioned earlier, we consider the three observation models corresponding to $\theta = 0$, 0.5 and 1. In each case, we use two sample sizes $M = 100$ and $M = 20$. To evaluate the performance of VEnKF, we implement both the AM based and the SVGD based VEnKF algorithms. As a comparison, we also impliment several commonly used methods: the EnKF variant provided in Sect. 3.2, PF, and NLEAF (Lei and Bickel 2011) with first-order (denoted as NLEAF 1) and second-order (denoted as NLEAF 2) correction, in the numerical tests. The stopping criterion in AM-VEnKF is specified by $\Delta_k = 20$, $\Delta_F = 0.1$ and $K_{\max} = 1000$, while the step size ϵ_k in GD iteration is 0.001. In SVGD-VEnKF, the step size is also 0.001, and the stopping criterion is chosen in a way so that the number of iterations is approximately the same as that in AM-VEnKF. For the small sample size $M = 20$, in all the methods except PF, the sliding window localization [with $l = 3$ and $k = 2$; see Lei and Bickel (2011) for details] is used.

With each method, we compute the square of the estimator bias (i.e., the difference between the ensemble mean and the ground truth) at every time step

and then average the bias over the 40 different dimensions. The procedure is repeated 200 times for each method and all the results are averaged over the 200 trials to alleviate the statistical error.

The average bias for $\theta = 0$ is shown in Fig. 2 where it can be observed that in this case, while the other three

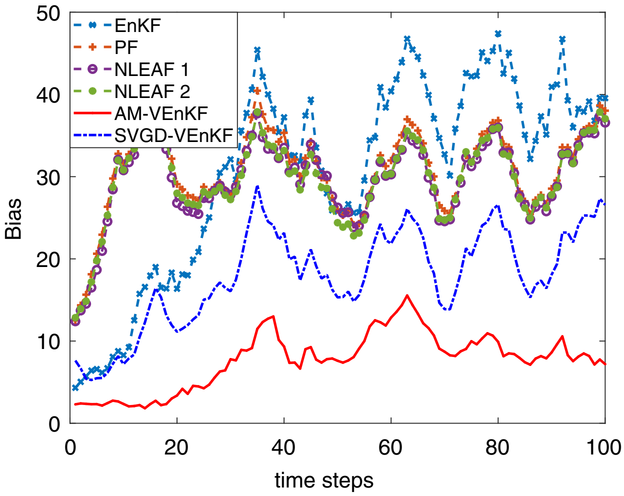


Fig. 2 The average bias at each time step for $\theta = 0$ and $M = 100$ in the Lorenz 96 example

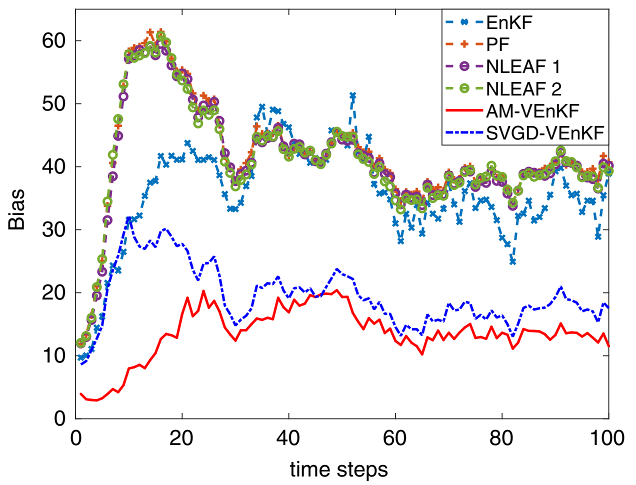


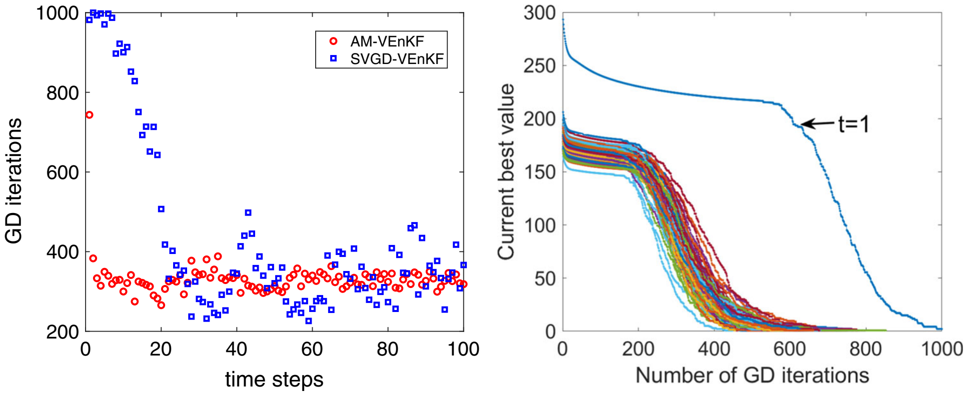
Fig. 4 The average bias at each time step for $\theta = 0.5$ and $M = 100$ in the Lorenz 96 example

methods yield largely comparable accuracy in terms of estimation bias, the bias of AM-VEnKF is significantly smaller. To analyze the convergence property of the method, in Fig. 3 (left) we show the number of GD iterations (of both AM and SVGD) at each time step, where one can see that all GD iterations terminate after around 300-400 steps in AM-VEnKF, except the iteration at $t = 1$ which proceeds for around 750 steps. The SVGD-VEnKF undergoes a much higher number of iterations in the first 20 time steps, while becoming about the same level as that of AM-VEnKF. This can be further understood by observing Fig. 3 (right) which shows the current best value F_k^* with respect to the GD iteration in AM-VEnKF, and each curve in the figure represents the result at a time step t. We see here that the current best values become settled after around 400 iterations at all time locations except $t = 1$, which agrees well with the number of iterations shown on the left. It is sensible that the GD algorithm takes substantially more iterations to converge at $t = 1$, as the posterior at $t = 1$ is typically much far away from the prior, compared to other time steps. These two figures thus show that the proposed stopping criteria are effective in this example.

The same sets of figures are also produced for $\theta = 0.5$ (Fig. 4 for the average bias and Fig. 5 for the number of iterations and the current best values) and for $\theta = 1$ (Fig. 6 for the average bias and Fig. 7 for the number of iterations and the current best values). Note that, in Fig. 6 the bias of EnKF is enormously higher than those of the other methods and so is omitted. The conclusions drawn from these figures are largely the same as those for $\theta = 0$, where the key information is that VEnKF significantly outperforms the other methods in terms of estimation bias, and within VEnKF, the results of AM are better than those of SVGD. Regarding the number of GD iterations in AM-VEnKF, one can see that in these two cases (especially in $\theta = 1$) it takes evidently more GD iterations for the algorithm to converge, which we believe is due to the fact that the noise in these two cases are not additive and so the observation models deviate further away from the Gaussian-linear setting.

As has been mentioned, we also conduct the experiments for a smaller sample size $M = 20$ with localization

Fig. 3 Left: the number of GD iterations (in both AM and SVGD) at each time step. Right: the current best value plotted against the GD iterations (in AM) where each line represents a time step. The results are for $\theta = 0$ and $M = 100$ in the Lorenz 96 example



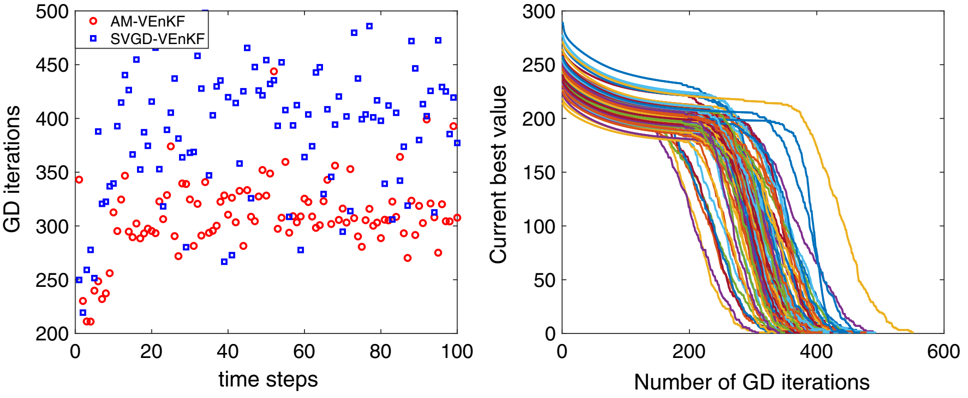


Fig. 5 Left: the number of GD iterations (in both AM and SVGD) at each time step. Right: the current best value plotted against the GD iterations (in AM) where each line represents a time step. The results are for $\theta = 0.5$ and $M = 100$ in the Lorenz 96 example

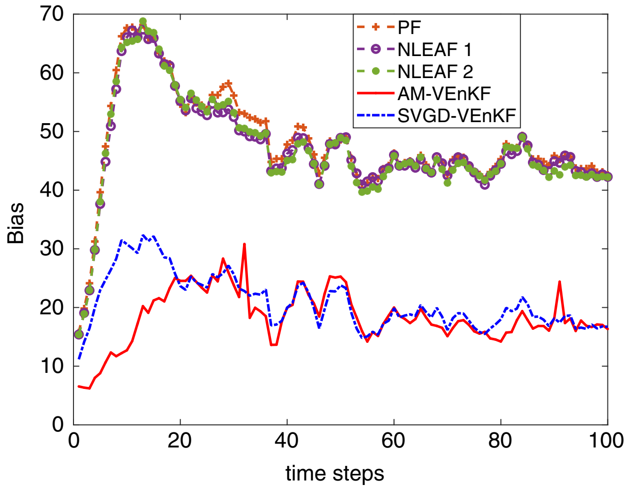


Fig. 6 The average bias at each time step for $\theta = 1$ and $M = 100$ in the Lorenz 96 example

employed, and we show the average bias results for $\theta = 0$, $\theta = 0.5$ and $\theta = 1$ in Fig. 8. Similar to the larger sample size case, the bias is also averaged over 200 trials. In this case, we see that the advantage of VEnKF is not as large as that for $M = 100$, but nevertheless VEnKF still yields clearly the lowest bias among all the tested methods. On the other hand, the results of the two VEnKF methods are quite similar while that of AM-VEnKF is slightly lower. Also shown in Fig. 8 are the number of GD iterations at each time step for all the three cases, which shows that the numbers of GD iterations used are smaller than their large sample size counterparts.

4.3 Fisher's equation

Our second example is the Fisher's equation, a baseline model of wildfire spreading, where filtering is often needed to assimilate observed data at selected locations into the model (Mandel et al. 2008). Specifically, the Fisher's equation is specified as follows,

$$c_t = Dc_{xx} + rc(1-c), \;\; 0 < x < L, \;\; t > 0, \tag{27a}$$

$$c_x(0,t) = 0, \;\; c_x(L,t) = 0, \;\; c(x,0) = f(x), \tag{27b}$$

where $D = 0.001$, $r = 0.1$, $L = 2$ are prescribed constants, and the noise-free initial condition $f(x)$ takes the form of,

$$f(x) = \begin{cases} 0, & 0 \le x < L/4 \\ 4x/L - 1, & L/4 \le x < L/2 \\ 3 - 4x/L, & L/2 \le x < 3L/4 \\ 0, & 3L/4 \le x \le L. \end{cases} \tag{28}$$

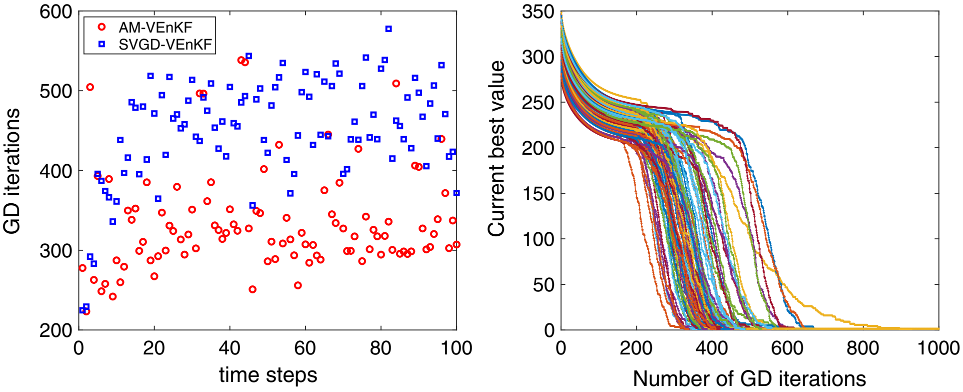


Fig. 7 Left: the number of GD iterations (in both AM and SVGD) at each time step. Right: the current best value plotted against the GD iterations (in AM) where each line represents a time step. The results are for $\theta = 1$ and $M = 100$ in the Lorenz 96 example

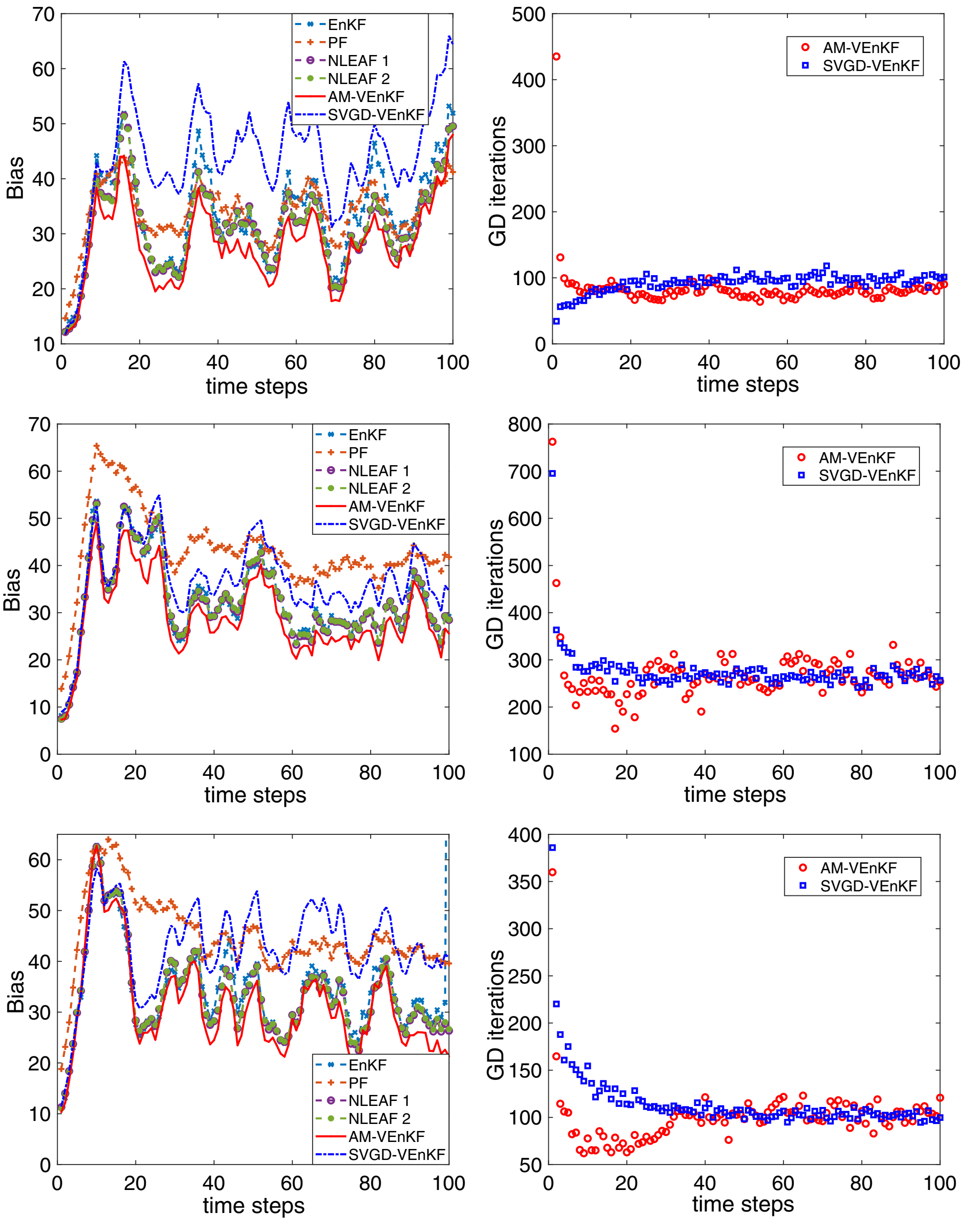


Fig. 8 The results for $M = 20$ in the Lorenz 96 example. The figures on the left show the average bias at each time step; the ones on the right show the number of GD iterations (in both AM and SVGD) at each time step. From top to bottom are respectively the results of $\theta = 0$, 0.5 and 1

In the numerical experiments we use an upwind finite difference scheme and discretize the equation onto $N_x = 200$ spatial grid points over the domain $[0, L]$, yielding a 200 dimensional filtering problem. The time step size is determined by $D\frac{\Delta t}{\Delta x^2} = 0.1$ with $\Delta x = \frac{L}{N_x - 1}$ and the total number of time steps is 60. The prior distribution for the initial condition is $U[-5, 5] + f(x)$, and in the numerical scheme a model noise is added in each time step and it is assumed to be in the form of $N(0, C)$, where

$$C(i, j) = 0.3 \exp(-(x_i - x_j)^2/L), \; i, \; j = 1, \ldots, N_x,$$

with x_i, x_j being the grid points.

The observation is made at each grid point, and the observation model is as described in Sect. 4.1. Once again we test the three cases associated with $\theta = 0$, 0.5 and 1. The ground truth and the data are both simulated from the model described above.

We test the same set of filtering methods as those in the first example. Since in practice, it is usually of more interest to consider a small ensemble size relative to the dimensionality, we choose to use 50 particles for this 200 dimensional example. Since the sample size is smaller than the dimensionality, the sliding window localization with $l = 5$ and $k = 3$ is used. All the simulations are repeated 200 times and the average biases are plotted in Fig. 9 for all the three cases ($\theta = 0$, 0.5 and 1). We see that in all the three cases the two VEnKF

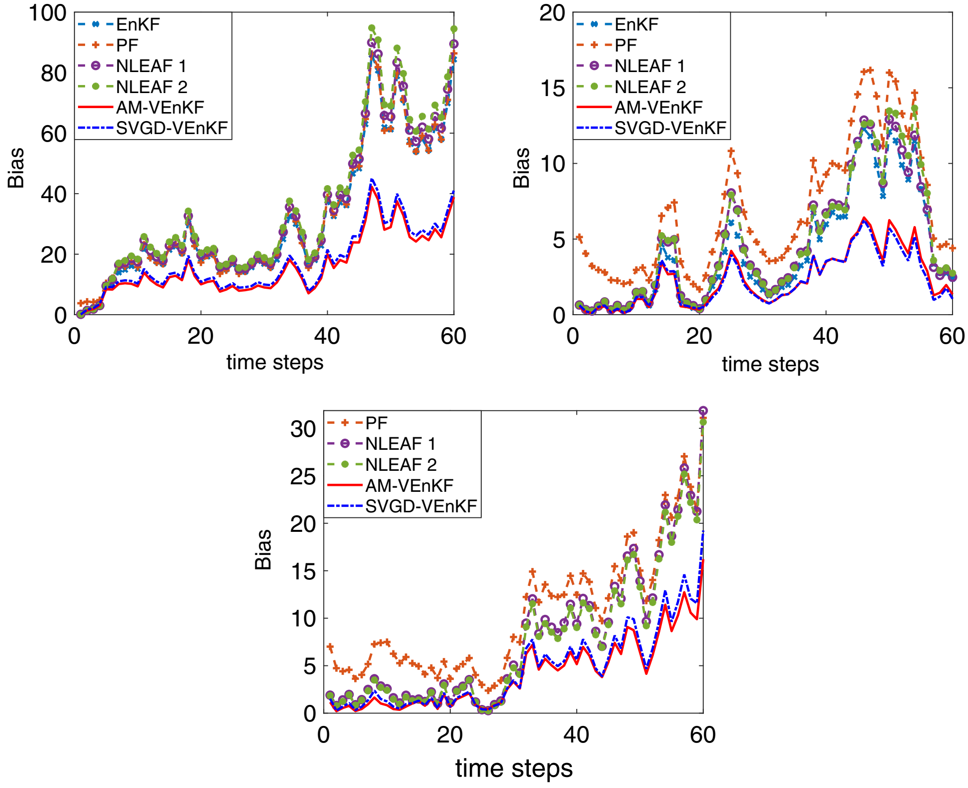


Fig. 9 The average bias at each time step in the Fisher's equation example. From top to bottom: $\theta = 0$, $\theta = 0.5$ and $\theta = 1$

methods result in the lowest estimation bias among all the methods tested, and the results of the two VEnKF methods are rather similar. It should be mentioned that, in the case of $\theta = 1$, the bias of EnKF is omitted as it is enormously higher than those of the other methods.

As the bias results shown in Fig. 9 are averaged over all the dimensions, it is also useful to examine the bias at each dimension. We therefore plot in Fig. 10 the bias of each grid point at three selected time steps $t = 10$, 30, and 60. The figures illustrate that, at all these time steps, the VEnKF methods yield substantially lower bias at the majority of the grid points, which is consistent with the average bias results shown in Fig. 9.

We also report that, the wall-clock time for solving the optimization problem in each time step in AM-VEnKF is approximately 2.0 s (on a personal computer with a 3.6GHz processor and 16GB RAM), indicating a modest computational cost in this 200 dimensional example.

4.4 Lorenz 2005 model

Here we consider the Lorenz 2005 model (Lorenz 2005) which products spatially more smoothed model trajectory than Lorenz 96. The Lorenz 2005 model is written in the following scheme,

$$\frac{dx^n}{dt} = [x,x]^{K,n} - X^n + F, \quad n = 1, \ldots, N. \tag{29}$$

where

$$[x,x]^{K,n} = \sum_{j=-J}^{J}{}' \sum_{i=-J}^{J}{}' (-x^{n-2K-i}x^{n-K-j} + x^{n-K+j-i}x^{n+K+j})/K^2,$$

and this equation is composed with periodic boundary condition. F is the forcing term and K is the smoothing parameter while $K << N$, and one usually sets $J = \frac{K-1}{2}$ if K is odd, and $J = \frac{K}{2}$ if K is even. Noted that the symbol $\sum'$ denote a modified summation which is similarly with generally summation $\sum$ but the first and last term are divided by 2. Moreover if K is even the summation is $\sum'$, and if K is odd the summation is replaced by ordinary $\sum$.

It is worth noting that, when setting $K = 1$, $N = 40$, and $F = 8$, the model reduces to Lorenz 96. In this example, we set the model as $N = 560$, $F = 10$ and $K = 16$, resulting in a 560-dimensional filtering problem. Following the notations in Sect. 4.2, Lorenz 2005 is also represented by a standard discrete-time fourth-order Runge-Kutta solution of Eq. (29) with $\Delta t = 0.01$ where the same model noise is added, and the state and observation pair $\{\mathbf{x}_t, \mathbf{y}_t\}$ is similarly denoted by Eq. (26). We reinstate that in this example the observation model is chosen differently (see Sect. 4.1).

And the initial state is chosen to be $\mathbf{x}_0 \sim U[0, 5]$.

In this numerical experiments, we test the same set of methods as those in the first two examples, where in each

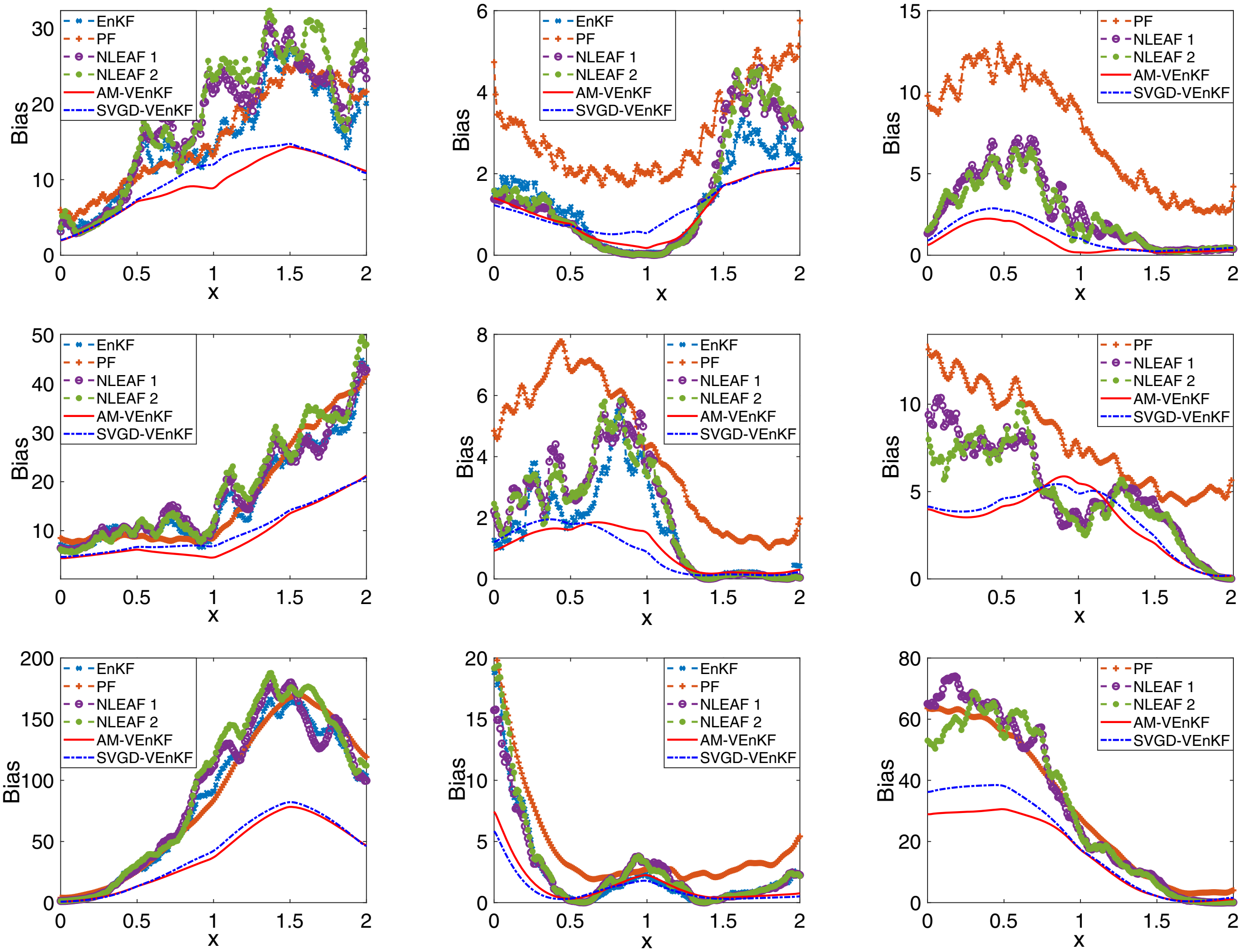


Fig. 10 The estimation bias at $t = 10$ (top), $t = 30$ (middle) and $t = 60$ (bottom), in the Fisher's equation example. From left to right: $\theta = 0$, $\theta = 0.5$ and $\theta = 1$

method 100 particles are used. Due to the small ensemble size, it is necessary to adopt the sliding-window localization with $(l, k) = (5, 3)$ in all methods except PF. We observe that the errors in the results of EnKF and PF are significantly larger than those in the other methods, and so those results are not presented here. It should be noted that the stopping threshold is as $\Delta_F = 0.5$ during nearest $\Delta_k = 20$ iterations in AM-VEnKF. All methods are repeated 20 times and we plot the averaged bias and the averaged GD iterations for all the three cases ($\epsilon = 0$, 0.5 and 1) in Fig. 11. One can see from the figures that, in the first case ($\epsilon = 0$) the results of all the methods are quite similar, while in the other two cases, the results of AM-VEnKF are clearly better than those of all the other methods.

5 Closing remarks

We conclude the paper with the following remarks on the proposed VEnKF framework. First

we reinstate that, the Fisher's equation example demonstrates that the KLD minimization problem in AM-VEnKF can be solved rather efficiently, and more importantly this optimization step does not involve simulating the underlying dynamical model. As a result, this step, though more complicated than the update in the standard EnKF, may not be the main contributor to the total computational burden, especially when the underlying dynamical model is computational intensive. Second, it is important to note that, although VEnKF can deal with generic observation models,

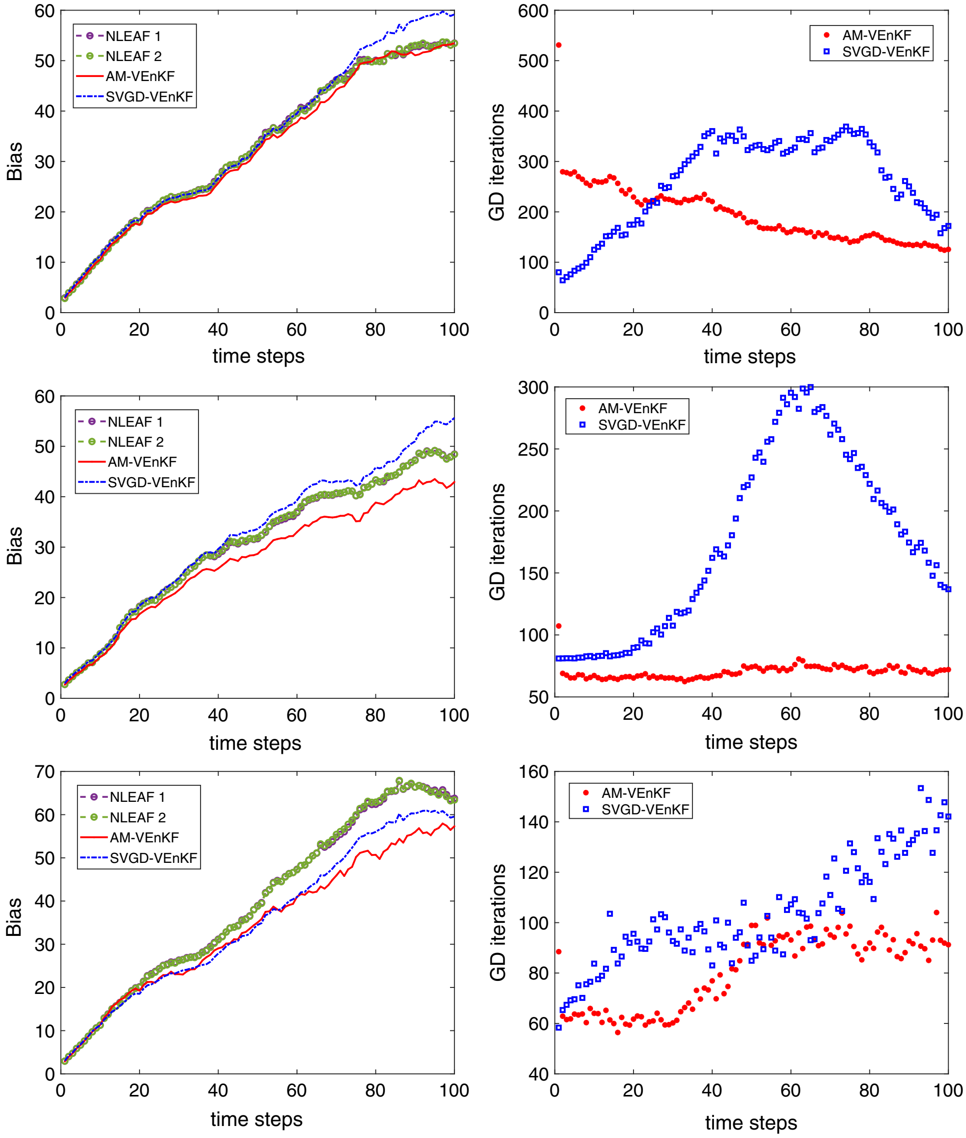


Fig. 11 The results for the Lorenz 2005 example: the figures on the left show the average bias at each time step; the ones on the right show the number of GD iterations (in both AM and SVGD) at each time step. From top to bottom are respectively the results of $\theta = 0$, 0.5 and 1

it still requires that the posterior distributions are reasonably close to Gaussian, an assumption needed for all EnKF type of methods. For strongly non-Gaussian posteriors, it is of our interest to explore the possibility of incorporating VEnKF with some existing extensions of EnKF that can handle strong non-Gaussianity, such as the mixture Kalman filter (Stordal et al. 2011). Finally, in this work we provide two transform mappings, the affine mapping and the RKHS mapping in the SVGD framework. In the numerical examples studied here, the affine mapping exhibits better performance, but we acknowledge that more comprehensive comparisons should be done to understand the advantages and limitations of different types of mappings. A related issue is that, some existing works such as Pulido and van Leeuwen (2019) use more flexible and complicated mappings and so that they can approximate arbitrary posterior distributions. It is worth noting, however, this type of methods are generally designed for problems where a rather large number of particles can be afforded, and therefore are not suitable for the problems considered here. Nevertheless, developing more flexible mapping based filters is an important topic that we plan to investigate in future studies.

Author Contributions L.W. and J.L. conceived of the presented idea, J.L. developed the theory and L.W. performed the computations. Both authors discussed the results and contributed to the final manuscript.

Declarations

Conflict of interest The authors declare no competing interests.

A SVGD-VEnKF

In this section, we discuss the procedure for constructing the mapping using the Stein variational gradient descent (SVGD) formulation (Liu and Wang 2016), which provides a nonlinear transform from the prior to the posterior in each time step.

Recall that in Sect. 3 we want to find a mapping by solving

$$\min_{T\in\mathcal{H}} \mathcal{D}_{\mathrm{KL}}(\pi_T, q), \tag{30}$$

where $q(\cdot) = \hat{\pi}(\cdot|y_{1:t})$ and $\mathcal{H}$ is a certain function space that will be specified later.

Following the same argument in Sect. 3.3, we obtain that Eq. (30) is equivalent to,

$$\min_{T\in\mathcal{H}} \mathcal{D}_{\mathrm{KL}}(p(\tilde{x}), q_{T^{-1}}(\tilde{x})), \tag{31}$$

where $q_{T^{-1}}(\cdot)$ is as defined in Sect. 3.3.

Now we need to determine the function space $\mathcal{H}$. While in the proposed AM-VEnKF method $\mathcal{H}$ is chosen to be an affine mapping space, the SVGD framework specifies $\mathcal{H}$ via a reproducing kernel Hilbert space (RKHS) (Scholkopf and Smola 2018).

First we write the mapping T in the form of,

$$T(\tilde{x}) = \tilde{x} + \tau\phi(\tilde{x}), \tag{32}$$

where τ is a prescribed stepsize.

Next we assume that mapping ϕ is chosen from a RKHS $\mathcal{H}_K$ specified by a reproducing kernel $K(\cdot,\cdot)$. Therefore the optimisation problem (31) becomes,

$$\min_{\phi\in\mathcal{H}_K} \mathcal{D}_{\mathrm{KL}}(p(\tilde{x}), q_{T^{-1}}(\tilde{x})). \tag{33}$$

In the SVGD framework, one does not seek to solve the optimisation problem in Eq. (33) directly; instead it can be derived that the direction of steepest descent is

$$\phi^*(\cdot) = \mathrm{E}_{\tilde{x}\sim p}[\nabla_{\tilde{x}} \log q(\tilde{x})K(\tilde{x},\cdot) + \nabla_{\tilde{x}} K(\tilde{x},\cdot)]. \tag{34}$$

It should be noted that we omit the detailed derivation of Eq. (34) here and interested readers may consult (Liu and Wang 2016) for such details. The obtained mapping ϕ^* is then applied to the samples which pushes them toward the target distribution. This procedure is repeated until certain stopping conditions are satisfied. The complete SVGD based VEnKF algorithm is given in Alg. 2. Finally we note that, in the numerical experiments we use the squared exponential kernel with bandwidth h:

$$K(x, x') = \exp(-\|x - x'\|_2^2/h),$$

where the implementation details can be found in Liu and Wang (2016).

Algorithm 2 SVGD based variational EnKF (SVGD-VEnKF)

```
– Prediction:
    – Let $\tilde{x}_t^m = F_t(x_{t-1}^m, \alpha_t^m), m = 1, 2, \ldots, M$;
    – Let $\hat{\pi}(\cdot|y_{1:t-1}) = N(\tilde{\mu}_t, \tilde{\Sigma}_t)$ where $\tilde{\mu}_t$ and $\hat{\Sigma}_t$ are computed
      using Eq. (9);
– Update:
    – Let $q(\cdot) = \hat{\pi}(\cdot|y_{1:t}) \propto \hat{\pi}(\cdot|y_{1:t-1})\pi(y_t|\cdot)$;
    – Repeat the following steps until the stopping conditions are
      satisfied;
        • Let
              $\tilde{\phi}^*(\cdot) = \frac{1}{M}\sum_{m=1}^{M}[\nabla_{\tilde{x}_t^m} \log q(\tilde{x}_t^m)K(\tilde{x}_t^m,\cdot) + \nabla_{\tilde{x}_t^m} K(\tilde{x}_t^m,\cdot)]$.
        • Let $\tilde{x}_t^m \leftarrow \tilde{x}_t^m + \tau\tilde{\phi}^*(\tilde{x}_t^m), m = 1, \ldots, M$.
    – Let $x_t^m = \tilde{x}_t^m$, for $m = 1, \ldots, M$.
```

References

Anderson, J.L.: An ensemble adjustment kalman filter for data assimilation. Mon. Weather Rev. **129**(12), 2884–2903 (2001)

Anderson, J.L.: A local least squares framework for ensemble filtering. Mon. Weather Rev. **131**(4), 634–642 (2003)

Anderson, J.L.: Exploring the need for localization in ensemble data assimilation using a hierarchical ensemble filter. Physica D **230**(1–2), 99–111 (2007)

Arulampalam, M.S., Maskell, S., Gordon, N.J., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans. Signal Process. **50**(2), 174–188 (2002)

Auvinen, H., Bardsley, J.M., Haario, H., Kauranne, T.: The variational kalman filter and an efficient implementation using limited memory BFGS. Int. J. Numer. Methods Fluids **64**(3), 314–335 (2010)

Ba, Y., Jiang, L., Na, O.: A two-stage ensemble kalman filter based on multiscale model reduction for inverse problems in time fractional diffusion-wave equations. J. Comput. Phys. **374**, 300–330 (2018)

Bauer, P., Thorpe, A., Brunet, G.: The quiet revolution of numerical weather prediction. Nature **525**(7567), 47–55 (2015)

Beal, M.J., Ghahramani, Z., Rasmussen, C.E.: The infinite hidden Markov model. Adv. Neural Inf. Process. Syst. **1**, 577–584 (2002)

Bishop, C.H., Etherton, B.J., Majumdar, S.J.: Adaptive sampling with the ensemble transform Kalman filter. Part I: theoretical aspects. Mon. Weather Rev. **129**(3), 420–436 (2001)

Capaldi, A., Behrend, S., Berman, B., Smith, J., Wright, J., Lloyd, A.L.: Parameter estimation and uncertainty quantication for an epidemic model. Math. Biosci. Eng. 553 (2012)

Chen, Z., et al.: Bayesian filtering: from Kalman filters to particle filters, and beyond. Statistics **182**(1), 1–69 (2003)

Doucet, A., Johansen, A.M.: A tutorial on particle filtering and smoothing: fifteen years later. Handb. Nonlinear Filter. **12**(656–704), 3 (2009)

El Moselhy, T.A., Marzouk, Y.M.: Bayesian inference with optimal maps. J. Comput. Phys. **231**(23), 7815–7850 (2012)

Evensen, G.: The ensemble Kalman filter: theoretical formulation and practical implementation. Ocean Dyn. **53**(4), 343–367 (2003)

Evensen, G.: Data Assimilation: The Ensemble Kalman Filter. Springer, Berlin (2009)

Fine, S., Singer, Y., Tishby, N.: The hierarchical hidden Markov model: analysis and applications. Mach. Learn. **32**(1), 41–62 (1998)

Frei, M., Künsch, H.R.: Bridging the ensemble kalman and particle filters. Biometrika **100**(4), 781–800 (2013)

Hoang, T.-V., Krumscheid, S., Matthies, H.G., Tempone, R.: Machine learning-based conditional mean filter: a generalization of the ensemble Kalman filter for nonlinear data assimilation. *arXiv preprint*arXiv:2106.07908, (2021)

Houtekamer, P.L., Mitchell, H.L.: Data assimilation using an ensemble Kalman filter technique. Mon. Weather Rev. **126**(3), 796–811 (1998)

Houtekamer, P.L., Mitchell, H.L.: A sequential ensemble kalman filter for atmospheric data assimilation. Mon. Weather Rev. **129**(1), 123–137 (2001)

Krogh, A., Larsson, B., Von Heijne, G., Sonnhammer, E.L.L.: Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. J. Mol. Biol. **305**(3), 567–580 (2001)

Law, K., Stuart, A., Zygalakis, K.: Data Assimilation, p. 214. Springer, Cham (2015)

Law, K.J.H., Tembine, H., Tempone, R.: Deterministic mean-field ensemble Kalman filtering. SIAM J. Sci. Comput. **38**(3), A1251–A1279 (2016)

Lei, J., Bickel, P.: A moment matching ensemble filter for nonlinear non-gaussian data assimilation. Mon. Weather Rev. **139**(12), 3964–3973 (2011)

Li, W., Rosenthal, W.S., Lin, G.: Trimmed ensemble kalman filter for nonlinear and non-gaussian data assimilation problems. arXiv:1808.05465 (2018)

Liu, Q., Wang, D.: Stein variational gradient descent: a general purpose Bayesian inference algorithm. In: Proceedings of the 30th International Conference on Neural Information Processing Systems, pp. 2378–2386 (2016)

Lorenz, E.N.: Predictability: a problem partly solved. In: Proceedings of Seminar on Predictability, vol. 1 (1996)

Lorenz, E.N.: Designing chaotic models. J. Atmospheric Sci. **62**(5), 1574–1587 (2005)

MacKay, D.J.C.: Information Theory, Inference and Learning Algorithms. Cambridge University Press, Cambridge (2003)

Mandel, J., Bennethum, L.S., Beezley, J.D., Coen, J.L., Douglas, M.K., Craig, C., Vodacek, A.: A wildland fire model with data assimilation. Math. Comput. Simul. **79**(3), 584–606 (2008)

Ott, E., Hunt, B.R., Szunyogh, I., Zimin, A.V., Kostelich, E.J., Corazza, M., Kalnay, E., Patil, D.J., Yorke, J.A.: A local ensemble kalman filter for atmospheric data assimilation. Tellus A Dyn. Meteorol. Oceanogr. **56**(5), 415–428 (2004)

Posner, E.: Random coding strategies for minimum entropy. IEEE Trans. Inf. Theory **21**(4), 388–391 (1975)

Pulido, M., van Leeuwen, P.J.: Sequential monte Carlo with kernel embedded mappings: the mapping particle filter. J. Comput. Phys. **396**, 400–415 (2019)

Roth, M., Özkan, E., Gustafsson, F.: A student's t filter for heavy tailed process and measurement noise. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 5770–5774 (2013)

Scholkopf, B., Smola, A.J.: Learning with kernels: support vector machines, regularization, optimization, and beyond. Adaptive Computation and Machine Learning series (2018)

Solonen, A., Haario, H., Hakkarainen, J., Auvinen, H., Amour, I., Kauranne, T.: Variational ensemble kalman filtering using limited memory BFGS. Electron. Trans. Numer. Anal. **39**, 271–285 (2012)

Spantini, A., Baptista, R., Marzouk, Y.: Coupling techniques for nonlinear ensemble filtering. arXiv:1907.00389 (2019)

Stordal, A.S., Karlsen, H.A., Nævdal, G., Skaug, H.J., Vallès, B.: Bridging the ensemble kalman filter and particle filters: the adaptive gaussian mixture filter. Comput. Geosci. **15**(2), 293–305 (2011)

Wainwright, M.J., Jordan, M.I., et al.: Graphical models, exponential families, and variational inference. Found. Trends® Mach. Learn. **1**(1–2), 1–305 (2008)

Whitaker, J.S., Hamill, T.M.: Ensemble data assimilation without perturbed observations. Mon. Weather Rev. **130**(7), 1913–1924 (2002)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.