Received: 15 December 2020 | Revised: 15 March 2021 | Accepted: 16 March 2021 | Published on: 5 May 2021

DOI: 10.1002/qj.4028

Quarterly Journal of the Royal Meteorological Society RMetS

RESEARCH ARTICLE

A particle flow filter for high-dimensional system applications

Chih-Chi Hu[1] | **Peter Jan van Leeuwen**[1,2]

[1]Department of Atmospheric Science, Colorado State University, Fort Collins, Colorado, USA

[2]Department of Meteorology, University of Reading, Reading, UK

Correspondence
Chih-Chi Hu, Department of Atmospheric Science, Colorado State University, Fort Collins, CO, 80526, USA.
Email: chihchi.hu@colostate.edu

Funding information
Cooperative Institute for Research of the Atmosphere; H2020 European Research Council, Grant/Award Number: 694509

Abstract

A novel particle filter proposed recently, the particle flow filter (PFF), avoids the long-existing weight degeneracy problem in particle filters and, therefore, has great potential to be applied in high-dimensional systems. The PFF adopts the idea of a particle flow, which sequentially pushes the particles from the prior to the posterior distribution, without changing the weight of each particle. The essence of the PFF is that it assumes the particle flow is embedded in a reproducing kernel Hilbert space, so that a practical solution for the particle flow is obtained. The particle flow is independent of the choice of kernel in the limit of an infinite number of particles. Given a finite number of particles, we have found that a scalar kernel fails in high-dimensional and sparsely observed settings. A new matrix-valued kernel is proposed that prevents the collapse of the marginal distribution of observed variables in a high-dimensional system. The performance of the PFF is tested and compared with a well-tuned local ensemble transform Kalman filter (LETKF) using the 1,000-dimensional Lorenz 96 model. It is shown that the PFF is comparable to the LETKF for linear observations, except that explicit covariance inflation is not necessary for the PFF. For nonlinear observations, the PFF outperforms LETKF and is able to capture the multimodal likelihood behavior, demonstrating that the PFF is a viable path to fully nonlinear geophysical data assimilation.

KEYWORDS

high-dimensional system, kernel embedding, non-Gaussian distribution, nonlinear data assimilation, particle filters, particle flows

1 | INTRODUCTION

The advancement of numerical weather prediction depends mainly on two factors: a model that can well depict the evolution of the system, and a desirable representation of the initial condition in the model. The importance of the latter can be significant if the underlying system is chaotic, in which case a small perturbation in the initial condition can grow rapidly in a short period of time. Data assimilation is a way to improve the initial state of the system.

To be more specific, the goal of the data assimilation is to sequentially estimate the probability of each possible model state given the information of the model forecast and observations. This means that the model state vector $\mathbf{x}$ (in $\mathbb{R}^{n_x}$, where n_x is the dimension of the model space) is

treated as a random vector. The goal is then to best describe the probability density function (pdf) of this random vector $\mathbf{x}$, given both the model forecast and the observations. We thus need to know how the forecast, or prior, pdf is updated with the information of the observations $\mathbf{y}$. In other words, we need to find the conditional probability of the model state given the knowledge of the observations. Using Bayes' theorem, the conditional probability $p(\mathbf{x}|\mathbf{y})$ can be written as

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})}{p(\mathbf{y})} p(\mathbf{x}) \tag{1}$$

where $p(\mathbf{x})$ is called the prior, $p(\mathbf{y}|\mathbf{x})$ is the likelihood, and $p(\mathbf{x}|\mathbf{y})$ is the posterior pdf.

To specify the prior, we need to evaluate the evolution of the pdf in the system. However, such evaluation of the pdf in a high-dimensional and nonlinear system is unachievable because of the high computational cost. A typical numerical weather prediction model has over 10^9 prognostic variables, and assimilates over 10^7 observations every 6–12 hr (Van Leeuwen *et al.*, 2019). A clever way to obtain the pdf is to adopt Monte Carlo methods. Monte Carlo methods are ways to approximate the evolution of a pdf given by the nonlinear system. First, randomly sample the initial pdf, and let the nonlinear system propagate the state of each sample. Then, use the subsequent distribution of the samples as an approximation of the prior pdf. However, it is difficult to infer the mathematical form of the pdf given a finite number of samples. An easy and useful way is to assume the pdf is Gaussian, and infer the statistics (i.e., mean and covariance) from the finite samples, as routinely done in ensemble Kalman filters (EnKF, Evensen, 1994; Houtekamer and Mitchell, 1998; Burgers *et al.*, 1998; see also Van Leeuwen, 2020 for a modern interpretation of the original EnKF).

The likelihood contains the relation between model state and the observations. If the observation errors are assumed to be Gaussian, as can often be done, and if the observation operator that relates the model state to the observations is linear, then the likelihood is Gaussian. Combined with a Gaussian prior, this leads to a Gaussian posterior, and its mean and covariance can be derived analytically; see, for example, the Kalman filter. However, when the observation is nonlinearly related to the model state (for example, in the case of satellite radiance observations), the Gaussian assumption for the likelihood fails.

One way to relax the Gaussian assumption on the likelihood is to use iterative methods such as a variational method. Variational methods seek the most likely model state in the posterior pdf by directly setting the gradient of the logarithm of the posterior pdf equal to zero. However, the underlying Gauss–Newton iterations can only find the solution closest to the first guess state, often taken as the prior mean, which is not the optimal solution if the posterior is multimodal. In addition, variational methods do not provide complete error estimates of the posterior, although they can be used to obtain an approximation of the inverse of the Hessian at the local mode. Hybrid methods, which combine ideas from ensemble Kalman filters and variational methods, suffer from similar problems when the posterior pdf is multimodal.

Although the posterior pdf can often be expected to be unimodal, for example, at the large scales in atmospheric models, related to the enormous amount of observations, it is well known that it can be multimodal at smaller scales. While recent studies have shown successes for data assimilation experiments using the EnKF or hybrids with the variational method even in convective scales (e.g., Snyder and Zhang, 2003; Zhang *et al.*, 2009; Jones *et al.*, 2016), many of them need several ad hoc settings, such as different ways to do the localization and the inflation to tackle the inappropriate assumptions. Some studies have come up with some clever ways to adaptively deal with these ad hoc settings (e.g., Zhang *et al.*, 2004; Anderson, 2007; Whitaker and Hamill, 2012; Ying and Zhang, 2015; Minamide and Zhang, 2019). Nevertheless, it is unclear whether the existing EnKF and variational methods based on the inappropriate assumptions are optimal. In other words, it is suggested that improvements on existing methods are in high demand in many geoscience data assimilation applications, and indeed possible.

One method that is not limited by any constraints in the prior distribution and linearity assumptions is the particle filter. It has the potential to better describe the full posterior pdf. However, the standard particle filter is known to suffer from the problem of weight degeneracy in high-dimensional problems (e.g., see Snyder *et al.*, 2008; Van Leeuwen, 2009 for details). This problem has made the application of particle filters in geoscience models difficult. Recently, much progress has been made to deal with the weight degeneracy problems, including the usage of the proposal density (e.g., Van Leeuwen, 2010), the introduction of localization in particle filters (e.g., Bengtsson *et al.*, 2003; Van Leeuwen, 2003; Poterjoy, 2016; Poterjoy *et al.*, 2019), and methods that try to transform the variables to Gaussian variables (Chorin *et al.*, 2010; Morzfeld *et al.*, 2012). See Van Leeuwen *et al.*, 2019 for a detailed review of these methods.

The so-called particle flow filter (Daum and Huang, 2011), a relatively new development in particle filtering, keeps all the particles at equal weight all the time. The particles are iteratively transformed from the prior to the posterior in state space, without the need to resample or to reweight the particles. Liu and Wang (2016) developed a static variant that is applicable

to high-dimensional spaces by embedding the transformation in a reproducing kernel Hilbert space. The sequential version of the Liu and Wang (2016) algorithm was developed by Pulido and van Leeuwen (2019). Pulido *et al.* (2019) successfully assimilated nonlinear observations, or observations with bimodal likelihood in the 40-variable Lorenz 1996 system. Since the PFF avoids the weight degeneracy problem, it has the potential to be applied in a high-dimensional system.

The motivation of this study is to investigate how to apply the PFF to high-dimensional nonlinear problems, and to compare the performance of the PFF with the local ensemble transform Kalman filter (LETKF), which has been widely used in geoscience applications. Special emphasis will be given to the formulation of the kernel, and it is shown that the scalar kernel used in earlier versions is insufficient in high-dimensional sparsely observed settings. A solution is found in matrix-valued kernels. We also discuss ways to formulate the prior from the forecast particles, another important ingredient in the methodology.

The remainder of this paper is organized as follows: Section 2 introduces the theoretical background for the PFF, and the effect of the different kernels will be discussed. Section 3 describes the data assimilation experiments used to evaluate the performance of the LETKF and the PFF. Section 4 compares the experiment results between a well-tuned LETKF and the PFF with linear and nonlinear observations. Section 5 conducts sensitivity experiments to see the effect of different settings in the PFF. Section 6 summarizes the results and discusses future applications to geosciences models.

2 | METHODOLOGY

2.1 | Introduction to the particle flow filter (PFF)

The particle flow filter (PFF, Pulido and van Leeuwen, 2019) iteratively transforms the particles from the prior to the posterior, with all their weights unchanged. In fact, weights play no role in this methodology. Specifically, the idea of particle flow is to (continuously) transform each state vector such that the pdf of the model state (recall that we treat the model state as a random vector) is transformed from the prior pdf to the posterior pdf. We define a pseudo time s, during which each state vector is transformed. We can then formulate the idea as

$$\frac{d}{ds}\mathbf{x}_s = \mathbf{f}_s(\mathbf{x}_s), \quad s \in [0, \infty]$$

$$q_0(\mathbf{x}) = p(\mathbf{x})$$
$$q_\infty(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}) \tag{2}$$

where $\mathbf{x}_s$ is a state at pseudo time s, $\mathbf{f}_s$ is the particle flow (the transformation) that can be evaluated at each state at pseudo time s, and q_s is the intermediate pdf of the state at pseudo time s: in particular, q_0 is the prior pdf, and q_∞ is the targeted pdf, which in our case is the posterior pdf.

The flow field $\mathbf{f}_s$ has to be chosen such that the distance between the pdf at pseudo time s (q_s) and the target pdf $q_\infty(\mathbf{x}) = p(\mathbf{x}|\mathbf{y})$ decreases as the pseudo time increases. Here, the Kullback–Leibler divergence (KL divergence) is used as a measure of this distance. (Formally the KL divergence is not a distance measure because it is not symmetric in its two arguments, but that is not a problem here.) Specifically, the KL divergence between the intermediate pdf at pseudo time s and the target pdf is

$$KL(q_s) = \int q_s(\mathbf{x}) \log\left(\frac{q_s(\mathbf{x})}{q_\infty(\mathbf{x})}\right) d\mathbf{x} \tag{3}$$

For a given targeted pdf, and a given prior from the forecast, the KL divergence is only a function of the intermediate pdf at each pseudo time s. To be efficient, the aim is to find the appropriate flow field $\mathbf{f}_s$ at each pseudo time such that the KL divergence can decrease as fast as possible.

There is an infinite number of choices for the flow field. To obtain a tractable solution, $\mathbf{f}_s$ is assumed to be in a reproducing kernel Hilbert space with a kernel $\mathbf{K}$, which is a mapping from $\mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbf{M}_{n_x \times n_x}(\mathbb{R})$, where n_x is the dimension of the system and $\mathbf{M}_{n_x \times n_x}(\mathbb{R})$ is a matrix of size n_x-by-n_x. By this assumption, we can derive the flow field $\mathbf{f}_s$ such that the KL divergence always decreases with respect to the pseudo time s (see Appendix for the derivations), as

$$\mathbf{f}_s(\cdot) = \mathbf{D} \int q_s(\mathbf{x})\{\mathbf{K}(\mathbf{x}, \cdot)\nabla_{\mathbf{x}} \log(p(\mathbf{x}|\mathbf{y})) + \nabla_{\mathbf{x}} \cdot \mathbf{K}(\mathbf{x}, \cdot)\} d\mathbf{x} \tag{4}$$

where $\mathbf{D}$ is a positive-definite matrix that we can choose. The matrix $\mathbf{D}$ is restricted in that it should ensure that the physical dimensions of $\mathbf{f}_s$ are those of the state vector, and we choose it equal to the localized prior covariance matrix. The Monte Carlo method is applied in the PFF. Denote the state of particles as $\mathbf{x}_s^{1:N_p}$, where the superscript is the index for particles and the subscript is the pseudo time. We use the particle representation for the intermediate pdf $q_s(\mathbf{x})$, as shown in Equation (5):

$$q_s(\mathbf{x}) = \frac{1}{N_p}\sum_{i=1}^{N_p} \delta(\mathbf{x} - \mathbf{x}_s^i) \tag{5}$$

and, therefore, the flow field at pseudo time s in Equation (4) can be written as

$$\mathbf{f}_s(\mathbf{x}) = \frac{1}{N_p}\mathbf{D}\sum_{i=1}^{N_p}\{\mathbf{K}(\mathbf{x}_s^i,\mathbf{x})\nabla_{\mathbf{x}_s^i}\log p(\mathbf{x}_s^i|\mathbf{y}) + \nabla_{\mathbf{x}_s^i}\cdot\mathbf{K}(\mathbf{x}_s^i,\mathbf{x})\} \tag{6}$$

Note that we have replaced the dot in Equation (4) with the state $\mathbf{x}$ whose particle flow is being evaluated in Equation (6). This formula is general for evaluating the transformation for any state, despite that we only need to evaluate the transformation at $\mathbf{x}_s^{1:N_p}$ for the N_p particles. To implement the PFF, we can discretize Equation (2) in pseudo time, and use Equation (6) to evaluate the movement of any state at pseudo time s to pseudo time $s+\Delta s$:

$$\mathbf{x}_{s+\Delta s} = \mathbf{x}_s + \frac{\Delta s}{N_p}\mathbf{D}\sum_{i=1}^{N_p} \{\mathbf{K}(\mathbf{x}_s^i,\mathbf{x}_s)\nabla_{\mathbf{x}_s^i} log\, p(\mathbf{x}_s^i|\mathbf{y}) + \nabla_{\mathbf{x}_s^i}\cdot\mathbf{K}(\mathbf{x}_s^i,\mathbf{x}_s)\} \tag{7}$$

with small Δs.

The first term on the right-hand side (RHS) of Equation (6) is an attracting term, which drives the state toward the local maximum of the posterior pdf, as the gradient of a function points toward its local maximum value. The kernel $\mathbf{K}(\mathbf{x}_s^i,\mathbf{x})$ measures how each particle $\mathbf{x}_s^{1:N_p}$ contributes to the local particle flow of the state $\mathbf{x}$ that is being evaluated. Therefore, the kernel in the first term acts as a weighting coefficient for each $\nabla_{\mathbf{x}_s^i} log p(\mathbf{x}_s^i|\mathbf{y})$. Note that the kernel should give larger weighting to those particles that are close to $\mathbf{x}$ to get an accurate average gradient of the posterior for state $\mathbf{x}$. In other words, the first term is the smoothed gradient at the state $\mathbf{x}$ represented by the neighboring particles. Figure 1a shows an example of how the kernel works.

The second term on the RHS of Equation (6) acts as a repelling term. To demonstrate its repelling nature, assume we have a scalar kernel:

$$\mathbf{K}(\mathbf{x}_s^i,\mathbf{x}) = K(\mathbf{x}_s^i,\mathbf{x})\mathbf{I}_{n_x} \tag{8}$$

where K is a function from $\mathbb{R}^{n_x}\times\mathbb{R}^{n_x}\to\mathbb{R}$, for instance:

$$K(\mathbf{x}_s^i,\mathbf{x}) = exp\left(-\frac{1}{2}(\mathbf{x}_s^i-\mathbf{x})^T\mathbf{A}(\mathbf{x}_s^i-\mathbf{x})\right) \tag{9}$$

Then, the divergence of the kernel is reduced to the gradient of the scalar function:

$$\nabla_{\mathbf{x}_s^i}\cdot\mathbf{K}(\mathbf{x}_s^i,\mathbf{x}) = \begin{bmatrix} \frac{\partial}{\partial x_1^i}K(\mathbf{x}_s^i,\mathbf{x}) \\ \frac{\partial}{\partial x_2^i}K(\mathbf{x}_s^i,\mathbf{x}) \\ \vdots \\ \frac{\partial}{\partial x_{n_x}^i}K(\mathbf{x}_s^i,\mathbf{x}) \end{bmatrix} = \nabla_{\mathbf{x}_s^i}K(\mathbf{x}_s^i,\mathbf{x}) \tag{10}$$

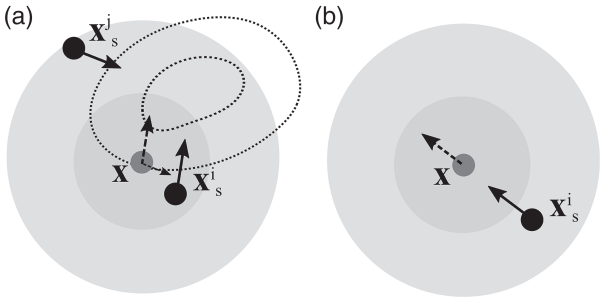

FIGURE 1 Demonstration of how the particle flow is represented by particles. The black dots are the states of particles, and the gray dot is the state $\mathbf{x}$ at which the particle flow is being evaluated. The shaded gray circles are the kernel, with darker shading representing greater value. (a) The weighted average of the gradient of the logarithm of posterior (the first term in Equation (6)). The dashed contours denote the posterior pdf. The solid arrows are the gradients at particles $\mathbf{x}_s^i$ and $\mathbf{x}_s^j$, which point toward the local maximum of the posterior pdf. The dashed arrow is the weighted average of the two solid arrows. Since $\mathbf{x}_s^i$ has a larger kernel value, the particle flow evaluated at $\mathbf{x}$ has a larger portion from $\mathbf{x}_s^i$ than from $\mathbf{x}_s^j$. (b) The divergence from the kernel (the second term in Equation (6)). The solid arrow is the direction of the gradient of the scalar kernel evaluated at the particle $\mathbf{x}_s^i$, while this flow is actually acting on the state $\mathbf{x}$, which is shown by the dashed arrow

Figure 1b shows an example. The gradient of the kernel should be evaluated at the particle positions $\mathbf{x}_s^i$ (i.e., $\mathbf{x}$ is fixed), and hence points from the particle $\mathbf{x}_s^i$ toward the state being evaluated $\mathbf{x}$, while this "force" is actually acting on the state being evaluated $\mathbf{x}$. That is, the gradient of the kernel tends to separate the state being evaluated $\mathbf{x}$ from each of the particles $\mathbf{x}_s^i$. Note that this behavior is true for any form of the kernel that maximizes when its two arguments are the same.

We can summarize the effect of each term in the particle flow as follows. Suppose that the posterior is a unimodal distribution; then the first term in the particle flow tends to make all the particles collapse into the mode, which is similar to the variational method in finding the most likely state. The second term tends to separate the particles from each other. When all the particles reach a steady state between these two forces, the distribution of the particles will follow the posterior pdf.

When evaluating Equation (6), the gradients of the logarithm of posterior (see Section 2.2) and of the kernel (see Section 2.3) need to be determined, which will be discussed in the following sections.

2.2 | The logarithm of the posterior in the PFF

The gradient of the logarithm of the posterior can be calculated analytically if we specify the form of the prior

and the likelihood. This is exactly the same as in a variational method. Specifically, based on Bayes' theorem (Equation (1)), we have

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y}) = \nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x}) + \nabla_{\mathbf{x}} \log p(\mathbf{x}) \tag{11}$$

To evaluate the gradient of the logarithm of the likelihood, we could, for instance, assume a Gaussian observational error, leading to

$$\log p(\mathbf{y}|\mathbf{x}) \propto -\frac{1}{2}\|\mathbf{y} - H(\mathbf{x})\|_{\mathbf{R}} \tag{12}$$

where $\|\mathbf{y} - H(\mathbf{x})\|_{\mathbf{R}} = (\mathbf{y} - H(\mathbf{x}))^T \mathbf{R}^{-1} (\mathbf{y} - H(\mathbf{x}))$, H is the observation operator, and $\mathbf{R}$ is the observation error covariance. Therefore, the gradient of the logarithm of likelihood can be obtained analytically,

$$\nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x}) = \mathbf{H}^T \mathbf{R}^{-1} (\mathbf{y} - H(\mathbf{x})) \tag{13}$$

where $\mathbf{H}$ is the linearized observation operator:

$$\mathbf{H} \coloneqq \frac{dH}{d\mathbf{x}}(\mathbf{x}) \tag{14}$$

Note that the observation error distribution does not have to be Gaussian in the PFF. This is a strong point of the PFF since the observation error may be non-Gaussian, for instance, when the observation operator H has a complex representation error.

When the observation operator H is linear, the linearized observation operator $\mathbf{H}$ is independent of the state $\mathbf{x}$. In the EnKF, the observation operator H is typically evaluated at each ensemble member, after which covariance is determined between state and observation space. This covariance is not state dependent, which is not optimal if H is highly nonlinear. The advantage of the PFF is that it can evaluate $\mathbf{H}$ locally for each ensemble member, which gives the PFF great potential to be applied to nonlinear problems.

For the gradient of the prior, we can for instance assume the prior to be Gaussian distributed, so that the gradient can be obtained analytically as

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}) = \mathbf{B}^{-1} (\mathbf{x} - \mathbf{x_b}) \tag{15}$$

where $\mathbf{x_b}$ and $\mathbf{B}$ are the prior mean and covariance matrix, respectively. Note that also the prior does not have to be the Gaussian in the PFF. Any form of pdf can be used for the prior in PFF as long as the gradient of its logarithm can be easily obtained.

2.3 | The choice of kernel

The solution of the PFF is independent of the choice of the kernel for an infinite ensemble size (Lu *et al.*, 2019). However, with only a finite number of particles, the particle distribution may not be unique. In other words, given different kernels (i.e., resulting in different particle flows), the final position of the particles in state space will be different, but their statistics should represent the posterior pdf as accurately as possible.

When the PFF was first developed, the kernel was chosen to be diagonal and isotropic, that is,

$$\mathbf{K}(\mathbf{x}, \mathbf{z}) = K(\mathbf{x}, \mathbf{z}) \mathbf{I}_{n_x} \tag{16}$$

where K is a function from $\mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}$:

$$K(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{z})^T \mathbf{A} (\mathbf{x} - \mathbf{z})\right) \tag{17}$$

and $\mathbf{A}$ is a matrix that can properly define the distance between particles in space. For example, we can choose $\mathbf{A}$ to be proportional to the inverse of the prior covariance:

$$\mathbf{A} = (\alpha \mathbf{B})^{-1} \tag{18}$$

To contrast with a kernel that will be introduced later, we will refer to the kernel in Equation (16) as the scalar kernel. The divergence of the scalar kernel is

$$\nabla_{\mathbf{x}} \cdot \mathbf{K}(\mathbf{x}, \mathbf{z}) = \begin{bmatrix} \frac{\partial}{\partial x_1} K(\mathbf{x}, \mathbf{z}) \\ \frac{\partial}{\partial x_2} K(\mathbf{x}, \mathbf{z}) \\ \vdots \\ \frac{\partial}{\partial x_{n_x}} K(\mathbf{x}, \mathbf{z}) \end{bmatrix} = -\mathbf{A}^T (\mathbf{x} - \mathbf{z}) K(\mathbf{x}, \mathbf{z}) \tag{19}$$

Pulido and van Leeuwen (2019) have shown that the scalar kernel works well for the 40-variable Lorenz 96 system with 20 observations (50% of the system is observed). Lu *et al.* (2019) has shown what any symmetric smooth kernel will do when the ensemble size is infinitely large, but when the ensemble size is much smaller than the dimension of the system, care needs to be taken in formulating the kernel. When we extend the system to 1,000 variables with 250 observations (25% of the system is observed), problems arise when using the scalar kernel, no matter the specific shape of K.

In this study, we generalize the scalar kernel to the matrix-valued kernel with

$$\mathbf{K}(\mathbf{x}, \mathbf{z}) = \mathrm{diag}([K_{(1)}(\mathbf{x}, \mathbf{z}), K_{(2)}(\mathbf{x}, \mathbf{z}), \ \dots \ , K_{(n_x)}(\mathbf{x}, \mathbf{z})]) \tag{20}$$

where

$$K_{(a)}(\mathbf{x}, \mathbf{z}) = K_{(a)}(x_{(a)}, z_{(a)}) = \exp\left(-\frac{1}{2}\frac{(x_{(a)} - z_{(a)})^2}{\alpha \sigma_{(a)}{}^2}\right) \tag{21}$$

where $x_{(a)}$ is the a-th component of the vector $\mathbf{x}$, that is,

$$\mathbf{x} = \begin{bmatrix} x_{(1)} \\ x_{(2)} \\ \vdots \\ x_{(n_x)} \end{bmatrix} \tag{22}$$

and $\sigma_{(a)}$ is the *SD* of the a-th component for the prior, and α is a tunable multiplication factor determining the width of kernel, which is chosen to be the reciprocal of the number of particles. The sensitivity of the PFF to the kernel width α will be discussed in Section 5. The divergence of the matrix-valued kernel is

$$\begin{aligned} \nabla_{\mathbf{x}_s^i} \cdot \mathbf{K}(\mathbf{x}_s^i, \mathbf{x}) &= \begin{bmatrix} \frac{\partial}{\partial x_1^i} K_{(1)}(x_{s,(1)}^i, x_{(1)}) \\ \frac{\partial}{\partial x_2^i} K_{(2)}(x_{s,(2)}^i, x_{(2)}) \\ \vdots \\ \frac{\partial}{\partial x_{n_x}^i} K_{(n_x)}(x_{s,(n_x)}^i, x_{(n_x)}) \end{bmatrix} \\ &= \begin{bmatrix} -\left(\frac{x_{s,(1)}^i - x_{(1)}}{\alpha \sigma_{(1)}^2}\right) K_{(1)}(x_{s,(1)}^i, x_{(1)}) \\ -\left(\frac{x_{s,(2)}^i - x_{(2)}}{\alpha \sigma_{(2)}^2}\right) K_{(2)}(x_{s,(2)}^i, x_{(2)}) \\ \vdots \\ -\left(\frac{x_{s,(n_x)}^i - x_{(n_x)}}{\alpha \sigma_{(n_x)}^2}\right) K_{(n_x)}(x_{s,(n_x)}^i, x_{(n_x)}) \end{bmatrix} \end{aligned} \tag{23}$$

We note that the most important difference between the scalar (Equation (16)) and the matrix-valued (Equation (20)) kernel is that, in the scalar kernel, the value of kernel is the same for all components of the two given particles, while that is typically not true for the matrix-valued kernel. The value of the scalar kernel can be seen as a generalized distance between particles. In other words, we only measure a single distance in the whole state space between two particles using the scalar kernel, while we independently measure the distances in each component using the matrix-valued kernel. (That is, we obtain n_x distances between two particles in the matrix-valued kernel.)

The differences in particle flow from using these two kernels become apparent when the convergence rate between components of the state is very different, which is the case when we only partially observe the system. Specifically, the convergence rate for the observed components is usually much faster than that for the unobserved components. In this case, even though the particles become very close in the observed component, the distance between two particles in terms of the full states can still be large because of the unobserved components, and hence, the value of the kernel can still be small when using the scalar kernel. In contrast, when the matrix-valued kernel is used, each component in the state vector feels its closeness of the corresponding component in the other particle. Specifically, the observed components feel each other independent of the unobserved components, and the kernel value on the observed component can be large when the observed components are close.

In addition to the value of the kernel, the divergence of the kernel also makes the particle flows from two kernels different. Specifically, for the scalar kernel, the divergence of the kernel (Equation (19)) is proportional to the value of the kernel itself. Each component of the divergence of the kernel is scaled by the same kernel value. When the observed components of the two particles become very close, the value of the kernel can still be negligible because of the contribution of the unobserved components, as discussed in the last paragraph. This means that the magnitude of the divergence of the kernel is also negligible. Therefore, the repelling force is too weak to separate the particles away from each other in the observed component using the scalar kernel, leading to particles collapsing toward the mode in the observed component. On the other hand, for the matrix-valued kernel, the value of each component in the divergence of the kernel (Equation (23)) is only proportional to how distance between two particles on this component behaves, and hence, the repelling force on the observed components can be effective.

Figure 2 shows a two-dimensional example of the difference in behavior of the two kernels. Suppose x_1 is the unobserved component and x_2 is the observed component. When x_1 and x_2 converge at a similar rate so that two particles become close in both directions at the same time, the difference of the repelling force (i.e., the divergence of kernel) between the two kernels is small (Figure 2a,b). This can occur when the observation has a strong impact on (or is highly correlated with) the update of the unobserved variable. When the unobserved component converges much slower than the observed component in Figure 2c,d, the repelling force of the scalar kernel remains small even though the observed components are very close (Figure 2d), leading to collapse of the particles onto the posterior mode in that direction. However, the matrix-valued kernel with independent repelling force in each component leads to a strong repelling force on the observed components (Figure 2c), avoiding the collapse on the posterior mode. We note that the divergence of the kernel is not only dependent on the kernel value but also on the factor in front, but the value of the kernel part turns out to be most important.

We demonstrate the difference of the posterior solution using the scalar and matrix-valued kernel in the 1,000-dimensional Lorenz 96 system detailed in Section 3. Figure 3 shows the posterior marginal distribution of the

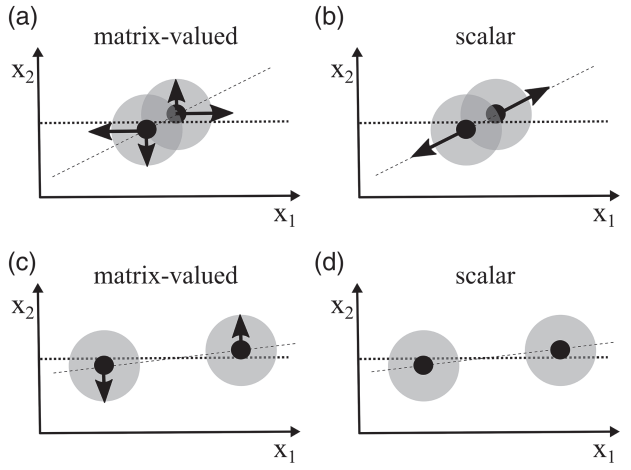

FIGURE 2 Comparison of the divergence of kernel between (a, c) the matrix-valued kernel and (b, d) the scalar kernel. The shading is the scalar kernel (which is assumed to be an independent Gaussian with equal variance), the black dots are the particles, and the black arrows are the repelling forces resulting from the divergence of kernel. x1 denotes an unobserved component, and x2 is an observed component. (a, b) Divergence for both kernels when the convergence rate of observed and unobserved components is similar, leading to similar repelling forces. (c, d) Divergence for both kernels when the convergence rate of the observed component is larger than that of the unobserved component. For the scalar kernel, the repelling force is small in all directions, such that the particles collapse to the posterior mode in the observed component. The matrix-valued kernel shows the correct behavior with a strong repellent force for the observed component because the divergence is allowed to be very different for the different components

variable $x_{(19)}$ (unobserved component) and $x_{(20)}$ (observed component) after the first data assimilation update. This is similar to the case in Figure 2c,d: the matrix-valued kernel is able to keep particles away from each other, preventing the collapse in the observed component (Figure 3a), while the particles collapse for $x_{(20)}$ using the scalar kernel (Figure 3b).

As a final note, the matrix-valued kernel can be more general than the diagonal version we explore here. Off-diagonal elements could be used to communicate the repelling force from one grid point to its neighbors. We did not need that in the experiments below, but it is an interesting path of research for the future.

2.4 | Implementation of the PFF

Algorithm 1 summarizes the steps for implementing the PFF. We can use the gradient descent method to push the particles so that each particle is moved along the direction of the steepest descent of the KL divergence. However, the convergence rate using the gradient descent is too slow. We use a quasi-Newton method with the prior covariance as preconditioner to speed up convergence (see, e.g., Nocedal and Wright, 2006). The calculation of the prior covariance is required in the PFF, so there is no additional cost if we choose the prior covariance as the preconditioner.

Another advantage of using the prior covariance as the preconditioner is that it helps to maintain the dynamical balance between the state variables. It is important in the numerical weather prediction model that the initial condition after the data assimilation update should stay in a physically realistic regime and "balanced," meaning that the update does not put the system in an unbalanced state resulting in a rapid transition immediately after the update, for example, by producing unrealistic gravity waves, which will affect the quality of the forecast.

Algorithm 1

The algorithm below is provided to aid implementation for low-to-moderate dimensional systems. For high-dimensional systems, modifications will be needed to make the implementation more efficient

[Input to the algorithm]
$\mathbf{x}_0^i$ (i = 1, … , N_p) [the prior ensemble] ($N_x \times 1$)
$\mathbf{y}$ [observation] ($N_y \times 1$)
$\mathbf{R}$ [observation error covariance] ($N_y \times N_y$)

[Assume a Gaussian prior in this pseudo code]
$\overline{\mathbf{x}}_0 \leftarrow \frac{1}{N_p}\sum_{i=1}^{N_p}\mathbf{x}_0^i$ [ensemble mean] ($N_x \times 1$)
$\mathbf{X} \leftarrow [\mathbf{x}_0^1 - \overline{\mathbf{x}}_0, \mathbf{x}_0^2 - \overline{\mathbf{x}}_0, \ldots, \mathbf{x}_0^{N_p} - \overline{\mathbf{x}}_0]$ [ensemble perturbation matrix] ($N_x \times N_p$)
$\mathbf{B} \leftarrow \frac{1}{N_p - 1}\mathbf{X}\mathbf{X}^T$ [prior covariance matrix] ($N_x \times N_x$)
$\mathbf{B} \leftarrow \mathbf{B} \circ \mathbf{C}$ [localization, see Equations (28) and (29)] ($N_x \times N_x$)

s = 0 [pseudo time for the data assimilation]
repeat
  for i = 1, … ,N_p.
    $\mathbf{y}^i \leftarrow H(\mathbf{x}_s^i)$ [modeled observation] ($N_y \times 1$)
    $\mathbf{H}^i \leftarrow \frac{dH}{d\mathbf{x}}(\mathbf{x}_s^i)$ [linearized observation operator] ($N_y \times N_x$)
    $\nabla logp(\mathbf{x}_s^i|\mathbf{y}) \leftarrow \mathbf{H}^{iT}\mathbf{R}^{-1}(\mathbf{y} - \mathbf{y}^i) - \mathbf{B}^{-1}(\mathbf{x}_s^i - \overline{\mathbf{x}}_0)$ [gradient of log posterior] ($N_x \times 1$)
  end for

  for $d = 1, \ldots, N_x$
    for $i = 1, \ldots, N_p$
      $\mathbf{f}_{s(d)}^i = 0$ [the (d)-th component of the particle flow for the i-th particle] (1×1)
      $x_{(d)}^i \leftarrow \mathbf{e}_{(d)}{}^T\mathbf{x}_s^i$ [the (d)-th component of $\mathbf{x}_s^i$] (1×1)
      $\frac{\partial p}{\partial x_{(d)}}^i \leftarrow \mathbf{e}_{(d)}{}^T\nabla logp(\mathbf{x}_s^i|\mathbf{y})$ [the (d)-th component of $\nabla logp(\mathbf{x}_s^i|\mathbf{y})$] ($1 \times 1$)

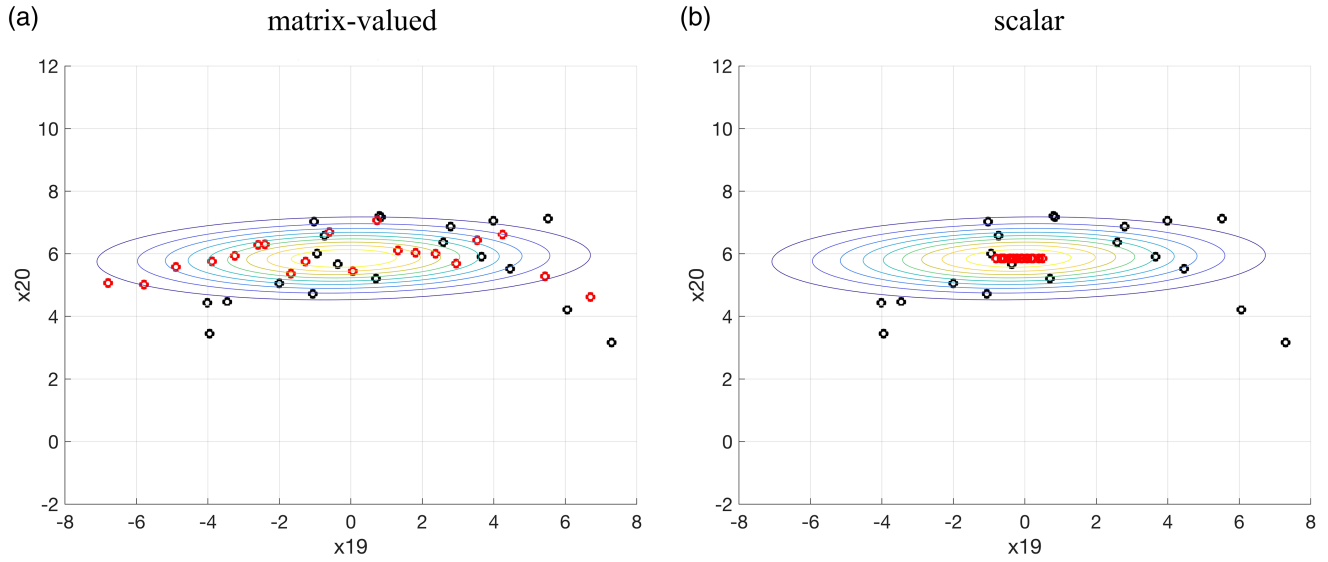

FIGURE 3 The prior and posterior marginal distribution of the variable $x_{(19)}$ (unobserved component) and $x_{(20)}$ (observed component) before and after the first data assimilation update ($t = 20$) with linear observation operator defined in Equation (30), using the (a) matrix-valued kernel (b) scalar kernel. The black circles are the particles for the prior, red circles are the particles for the posterior, and the contour is the posterior covariance given by the ensemble Kalman filter [Colour figure can be viewed at wileyonlinelibrary.com]

for j = 1, … ,N_p

$x^j_{(d)} \leftarrow \mathbf{e}_{(d)}{}^T \mathbf{x}^j_s$ [the (d)-th component of $\mathbf{x}^j_s$] (1×1)

$K^{i,j}_{(d)} \leftarrow exp\left(-\frac{1}{2}\frac{(x^i_{(d)}-x^j_{(d)})^2}{\alpha \mathbf{B}_{d,d}}\right)$ [the kernel Equation (21)] (1×1)

$\frac{\partial K}{\partial x}{}^{i,j}_{(d)} \leftarrow -\left(\frac{x^i_{(d)}-x^j_{(d)}}{\alpha \mathbf{B}_{d,d}}\right) K^{i,j}_{(d)}$ [the gradient of kernel Equation (23)] (1×1)

$\mathbf{I}_f{}^i_{(d)} \leftarrow \mathbf{I}_f{}^i_{(d)} + \frac{1}{N_p}\left(K^{i,j}_{(d)} \frac{\partial p}{\partial x}{}^i_{(d)} + \frac{\partial K}{\partial x}{}^{i,j}_{(d)}\right)$ [integral part of Equation (6)] (1×1)

end for

end for

end for

for i = 1, … ,N_p

$\mathbf{f}^i_s \leftarrow \mathbf{B}\mathbf{I}^i_f$ [multiply integral by $\mathbf{B}$ to find particle flow Equation (6)] $(N_x \times 1)$

$\mathbf{x}^i_s \leftarrow \mathbf{x}^i_s + \Delta s \mathbf{f}^i_s$ [Equation (7)] $(N_x \times 1)$

end for

$s \leftarrow s+1$

until stopping criterion met

[output of the algorithm]

$\mathbf{x}^i_s$ $(i = 1, \ldots N_p)$ [the posterior ensemble members] $(N_x \times 1)$

The pseudo time step (the time step for the iterations in Equation (7)) Δs should be small enough to prevent instability of the iterative procedure. We use an adaptive scheme to determine the pseudo time step Δs to accelerate the convergence. In general, we start with a small Δs at the beginning, and gradually increase Δs during the iterations. This is because at the beginning of the iterations the magnitude of the particle flow is large and we need a smaller Δs to ensure that the trajectories of the particles do not cross in state space. At later iterations, the magnitude of the particle flow becomes smaller, so we can use a larger Δs to accelerate the convergence. Specifically, we start with a small initial Δs based on trial and error, ensuring that the particle flow will not blow up in the first few iterations. Then, if the particle flow decreases for 20 pseudo time steps, we increase Δs by a factor of 1.4. If the magnitude of the particle flow increases, we will decrease Δs by a factor of 1.4. The initial Δs is chosen differently for different observation types. For the linear and absolute value observation, the initial Δs is 0.05, and for the square and exponential observation, the initial Δs is 0.001. More research is needed on how to accelerate the convergence in particle flows, but the practical scheme outlined above works well for our problem.

3 | EXPERIMENTAL DESIGN

We compare the effect of the PFF and the local ensemble transform Kalman filter (LETKF) by applying both methods to the Lorenz 96 model. The equation for the n_x-dimensional Lorenz 96 model is

$$\frac{dx_{(a)}}{dt} = (x_{(a+1)} - x_{(a-2)})x_{(a-1)} - x_{(a)} + F \qquad (24)$$

where a = 1, … , n_x and $x_{(a)}$ is the a-th component of the state $\mathbf{x}$ defined in Equation (22). We set n_x to 1,000 and F to 8. The fourth-order Runge–Kutta scheme is used, and the time resolution is $\Delta t = 0.01$. The initial condition is set as

$$x_{(a)}(t=0) = \begin{cases} F, & if\ mod(a,5) \quad \neq 0 \\ F+1 & if\ mod(a,5) \quad = 0 \end{cases} \tag{25}$$

and the model is integrated for 1,000 time steps to generate chaotic behavior. At t = 1,000, $N_p = 20$ ensemble members are generated by adding random perturbations, following a normal distribution $N(\mathbf{0}, 2\mathbf{I}_{n_x \times n_x})$. A run without the random perturbation at t = 1,000 is taken as the truth. After then, observations with random noise, following $N(\mathbf{0}, \mathbf{R})$, where $\mathbf{R} = \varepsilon \mathbf{I}_{n_y \times n_y}$ (ε will be different for different observation types) and n_y is the number of observations, are assimilated into the system every 20 time steps (which roughly corresponds to every 24 hr in atmospheric models). The magnitude of the observational error ε depends on the observation operators. The observation is taken at every fourth variable in the system, which means only 25% of the system is observed and $n_y = \frac{n_x}{4}$. Note that this is a fixed-observing system, meaning that 75% of the system is never observed. The ensemble and the truth are integrated for 1,500 time steps.

To obtain a more statistically reliable comparison between the PFF and the LETKF, the experiments described above are averaged over 10 different random realizations of the ensemble perturbations, the truth, and the observation errors for both the PFF and the LETKF. The only exception is for the square observation, in which case we are not able to find suitable parameters for the LETKF for 9 of the 10 realizations. In other words, the model blows up before 1,500 time steps for these nine realizations when using LETKF to assimilate the square observation. Nevertheless, the model remains stable using the PFF to assimilate square observations for all the 10 realizations. Therefore, when comparing the performance of the LETKF and the PFF for assimilating the square observation, the performance for the LETKF is evaluated by only one realization, while for the PFF is by all the 10 realizations. For other observations, we will compare the averaged performance of the results from all the 10 realizations in the following.

For both the PFF and the LETKF, we need to localize the effect of the observations due to the fact that the dimension of system is much larger than the sample size. For the LETKF, we update the system grid point by grid point, using the localized observational error covariance $\mathbf{R}_i$ when updating the i-th grid point:

$$\mathbf{R}_i = \mathbf{R} \circ \mathbf{C}_i \tag{26}$$

where $\circ$ is the Schur product of the observational error covariance and the matrix $\mathbf{C}_i$:

$$\mathbf{C}_i = \mathrm{diag}\left(\left[\exp\left\{-\left(\frac{d(i,1)}{r_{in}}\right)^2\right\}, \exp\left\{-\left(\frac{d(i,2)}{r_{in}}\right)^2\right\}, \dots, \exp\left\{-\left(\frac{d(i,n_y)}{r_{in}}\right)^2\right\}\right]\right) \tag{27}$$

where $d(i, j)$ is the distance between the i-th grid point and the j-th observation, and r_{in} is the decorrelation length scale for the observation, which is set to $r_{in} = 4$. The choice of the decorrelation length scale r_{in} is based on the properties of the Lorenz 96 system. We retain the covariance between the state variables that are within three times the decorrelation length scale from the observation location. That is, 25 ($= 1 + 2 \times 3 \times r_{in}$) state variables survive this localization.

For the PFF, we assume a Gaussian prior in the standard experiment described here. We directly localize on the prior covariance matrix:

$$\mathbf{B} \leftarrow \mathbf{B} \circ \mathbf{C} \tag{28}$$

where

$$\mathbf{C} = [c_{i,j}]_{n_x \times n_x}, c_{i,j} = exp\left\{-\left(\frac{i-j}{r_{in}}\right)^2\right\} \tag{29}$$

Note that r_{in} is also set as 4 here.

We compare the performance of the PFF and the LETKF using different types of observations, including linear and nonlinear observations. The linear observation operator is

$$H_{linear}(\mathbf{x}) = \begin{bmatrix} x_{(4)} \\ x_{(8)} \\ \vdots \\ x_{(n_x)} \end{bmatrix}_{n_y \times 1} \tag{30}$$

The observational error is set as $\varepsilon = 0.5$. For the nonlinear observations, we consider several observational operators: absolute value, exponential, and square operator. For the absolute value operator, which is

$$H_{abs}(\mathbf{x}) = \begin{bmatrix} |x_{(4)}| \\ |x_{(8)}| \\ \vdots \\ |x_{(n_x)}| \end{bmatrix}_{n_y \times 1} \tag{31}$$

the pdf of the likelihood as function of the state will be bimodal. The magnitude of the linearized observational operator will be independent of the state. The observational error is set as $\epsilon = 0.5$. We will also test the behavior of the methods for an exponential observation operator, given by

$$H_{exp}(\mathbf{x}) = \begin{bmatrix} e^{\frac{x_{(4)}}{6}} \\ e^{\frac{x_{(8)}}{6}} \\ \vdots \\ e^{\frac{x_{(n_x)}}{6}} \end{bmatrix}_{n_y \times 1} \tag{32}$$

leading to a unimodal likelihood. However, the magnitude of the linearized observational operator will depend on the state. The observational error is set as $\epsilon = 0.01$. Finally, for the squared operator, given by

$$H_{square}(\mathbf{x}) = \begin{bmatrix} {x_{(4)}}^2 \\ {x_{(8)}}^2 \\ \vdots \\ {x_{(n_x)}}^2 \end{bmatrix}_{n_y \times 1} \tag{33}$$

the pdf of the likelihood as function of the state will be bimodal, and the magnitude of the linearized observation operator will depend on the state, which is the most complicated operator. The observational error is set as $\epsilon = 1$.

4 | RESULTS

The performance of the PFF is tested in a sequential data assimilation experiment as described in Section 3. To compare the results between the PFF and the LETKF, the prior for the PFF is assumed to be Gaussian, which is the same assumption as is used in the LETKF. The difference between the PFF and the LETKF is then in the likelihood. When the observation is linear, their performance is expected to be similar. However, when the observation is nonlinear, the non-Gaussian likelihood is expected to cause differences in the two methods.

4.1 | Linear observation operator

To quantitatively compare the results from different data assimilation (DA) methods, the root mean square error (RMSE) of all the variables (total RMSE) is used to evaluate their performance:

$$RMSE\_X(t) = \sqrt{\frac{1}{n_x}\sum_{a=1}^{n_x}(\bar{x}_{(a)}(t) - x_{t(a)}(t))^2} \tag{34}$$

where $\bar{x}_{(a)}$ is the ensemble mean for the a-th component of the state vector $\bar{\mathbf{x}}$ and $x_{t(a)}$ is the a-th component of the truth. We also compare the RMSE of the observed variables alone for different experiments, which is

$$RMSE\_X\_OBS(t) = \sqrt{\frac{1}{|OBS|}\sum_{(a)\in OBS}(\bar{x}_{(a)}(t) - x_{t(a)}(t))^2} \tag{35}$$

where $OBS = \{4, 8, \ldots, n_x\}$ is the set containing the indices of the observed variables and $|OBS|$ is the number of the elements in the set OBS ($|OBS| = n_y = \frac{n_x}{4}$). Similarly, the RMSE of the unobserved variables is

$$RMSE\_X\_NOOBS(t) = \sqrt{\frac{1}{|NOOBS|}\sum_{(a)\in NOOBS}(\bar{x}_{(a)}(t) - x_{t(a)}(t))^2} \tag{36}$$

where $NOOBS = \{1, 2, \ldots, n_x\} \setminus OBS$ and $|NOOBS| = n_x - n_y = \frac{3}{4}n_x$.

Figure 4 shows the RMSE of the LETKF and the PFF results. Both methods show a reduced RMSE for both observed and unobserved variables compared with the ensemble without data assimilation (noDA ensemble). For the LETKF without inflation, Figure 4a shows that its RMSE of the observed variables decreases at the observation times. The RMSE of the unobserved variables also decreases at most of the observation times before $t = 800$, while it increases, instead, for most of the observation times after $t = 800$. This suggests that the covariance between observed and unobserved variables is not good enough to update the unobserved variables correctly after $t = 800$ for the LETKF without inflation. In addition, the RMSE of the observed variables at the observation times gradually increases with time. This suggests that the system is gradually biased against the observations.

After extensive experimentation, we found the best value for the inflation factor was 1.25 for the LETKF. Figure 4b shows that the RMSE of the observed variables decreases to around 0.6–0.7 at almost all the observation times. The RMSE of the unobserved variables also decreases for most of the observation times. This suggests that with the inflated prior, the covariance structure becomes better and LETKF is able to better follow the system without the possible filter divergence as in Figure 4a.

With a proper choice of the kernel width α in Equation (21), we are able to have a stable RMSE of the observed variables for the PFF either with or without inflation of the prior (Figure 4c,d). However, it is found that with the inflation of the prior in the PFF, the RMSE of both observed and unobserved variables still slightly decreases. Comparing the performance of the LETKF with

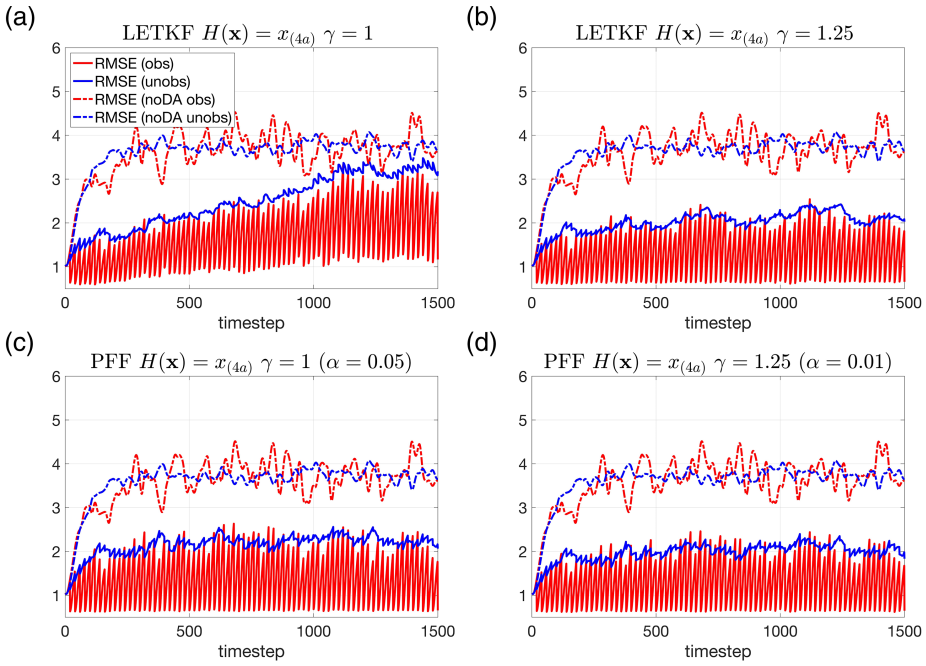

FIGURE 4 Root-mean-square error (RMSE) of the noDA ensemble (i.e., the ensemble with same initial conditions but without data assimilation, dashed lines) and with data assimilation (solid lines) using (a) the LETKF without inflation $\gamma = 1$, (b) the LETKF with inflation factor $\gamma = 1.25$, (c) the PFF without inflation $\gamma = 1$ and with kernel width $\alpha = 0.05$, and (d) the PFF with inflation factor $\gamma = 1.25$ and with kernel width $\alpha = 0.01$ [Colour figure can be viewed at wileyonlinelibrary.com]

inflated prior (Figure 4b) and the PFF with inflated prior (Figure 4d), we find their RMSE of the observed variables comparable, while it is interesting to note that the PFF has a slightly smaller RMSE of the unobserved variables than the LETKF does. Generally, Figure 4 suggests that the performance of the PFF is overall comparable to a well-tuned LETKF when the observation is linearly related to the model state.

In addition to the behavior of the mean, the "reliability" of the ensemble is also important. A "reliable ensemble" can defined as the ensemble in which "the truth and the forecast ensemble can be considered samples from the same probability distribution." (Hamill, 2001). In other words, the distribution represented by the ensemble is indistinguishable from the distribution from which the truth is drawn. To evaluate the reliability, the rank histogram is used as another measure of performance. A rank histogram close to a uniform distribution is considered as a necessary condition for a reliable ensemble.

The rank histogram of the observed variables in the prior compared with the truth at the observation times is shown in Figure 5. It is shown that the rank histogram for the LETKF without inflation (Figure 5a) is close to U-shape, suggesting that the ensemble may be either biased against the truth or underdispersive. Based on Figure 4a, since the RMSE of the observed variables at observation times gradually increases with time, it can be inferred that the ensemble is also gradually biased against the truth. With inflation of the prior, the rank histogram from the LETKF becomes flat (Figure 5b). The rank histogram from the PFF is quite close to the uniform distribution, despite that the ensemble is slightly overdispersive (Figure 5c,d). The rank histogram for the unobserved variables is similar to that of the observed variables (not shown).

In a brief summary, we find that, for the linear observation, the PFF shows comparable results to the LETKF with prior inflation. In the setup used here, both the PFF and the LETKF assume a Gaussian prior, and the observation operator is linear. In this case, we would expect that PFF shows similar results with a well-tuned LETKF. Note that both inflating the prior and tuning the kernel width (see Section 5.1) for the PFF can make the posterior wider, to prevent filter divergence. However, unlike the inflation for the prior, the tuning of kernel width will not change the position of the mean in the posterior. In other words, the tuning of the kernel width retains most of the information from both the prior and the observations. Note that when there is evidence that the prior is underdispersive, we can still inflate the prior for the PFF. This suggests that the PFF is more flexible than the LETKF.

4.2 | Nonlinear observation operators

When the model states are nonlinearly related to the observations, the likelihood is no longer Gaussian as a function of the model state, making the posterior non-Gaussian too. In this case, the mean of the ensemble model state may not be representative of the behavior of the ensemble, and so a RMSE in the model state may not be a useful measure of performance, for example, when the posterior is a multimodal distribution. This happens for the square observation $H(\mathbf{x}) = x_{(4a)}{}^2$, where the observation operator is not

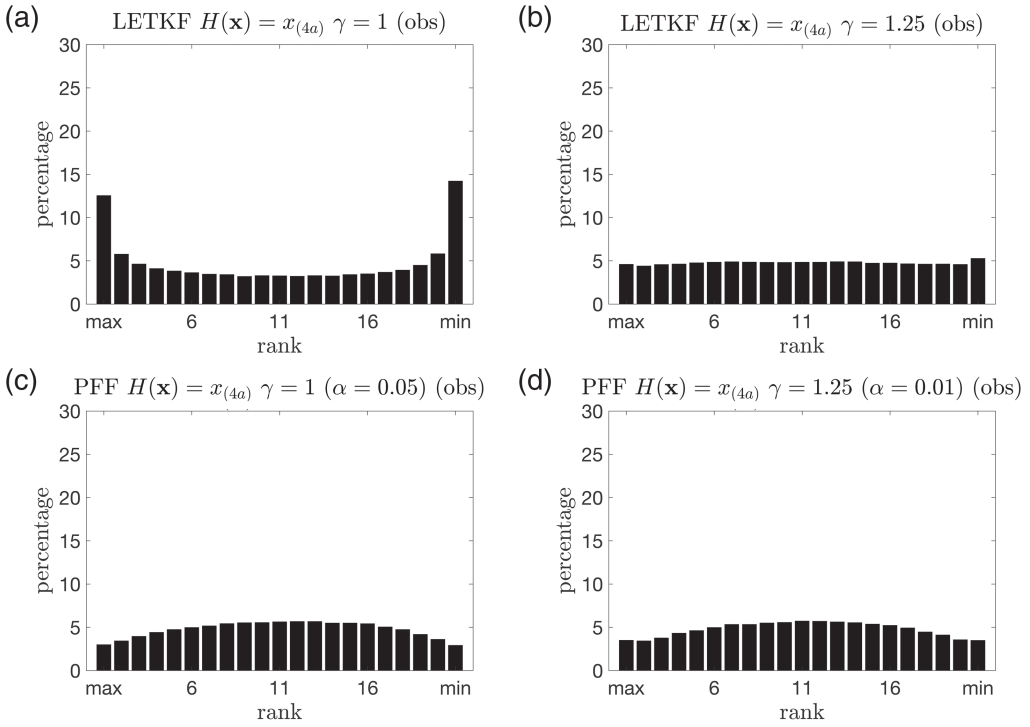

FIGURE 5 Rank histogram of the observed variables for the prior compared with the truth at the observation times for the linear observation. The ensemble from (a) the LETKF without inflation $\gamma = 1$, (b) the LETKF with inflation factor $\gamma = 1.25$, (c) the PFF without inflation $\gamma = 1$ and with kernel width $\alpha = 0.05$, and (d) the PFF with inflation factor $\gamma = 1.25$ and with kernel width $\alpha = 0.01$

one-to-one even if we confine the domain to the subset of the observed variable. To measure the performance of the ensemble for the observed variables, instead, we compare the RMSE defined in the observational space. Given that the measurement error (the observation error in the observational space) is taken to be Gaussian, the posterior in the observational space is expected to be closer to Gaussian, and at least less likely to be multimodal. The RMSE of the observed variables defined in the observational space is

$$RMSE\_Y(t) = \sqrt{\frac{1}{|OBS|}\sum_{(a)\in OBS}(\bar{y}_{(a)}(t) - y_{t(a)}(t))^2} \quad (37)$$

where $\bar{y}_{(a)}$ is the ensemble mean of the modeled observation given by the a-th component of the model state, as defined in Equation (38):

$$\bar{y}_{(a)} = \frac{1}{N_p}\sum_{i=1}^{N_p} H(x^i_{(a)}) \quad (38)$$

and $y_{t(a)}$ is the modeled observation given by the a-th component of the truth,

$$y_{t(a)} = H(x_{t,(a)}) \quad (39)$$

Figure 6 shows the RMSE and the spread of the observed variables in the observational space for the nonlinear observations. For the absolute value operator, both the RMSE and the spread at observation times gradually increase with time and become steady after $t = 300$ for the LETKF, and the LETKF can still slightly improve the ensemble compared with the noDA ensemble (Figure 6a). For the exponential operator, the results from LETKF are not very stable: the results are good before $t = 400$, but after $t = 400$, the error sometimes grows very fast between the observation times and sometimes is even larger than the noDA ensemble (Figure 6c). Similar results can be found for the square operator for the LETKF (Figure 6e). In contrast, the performance of the PFF for these three nonlinear observations are all very good and stable (Figure 6b,d,f). All of the three experiments show an improvement over the noDA ensemble. We note that the parameters (inflation factor and the localization radius) in LETKF for these experiments are already the best: if the parameters are changed even a small amount, the model blows up before $t = 1{,}500$.

For the behavior of the unobserved variables, the LETKF only shows an improvement in the RMSE for the absolute value observation before $t = 300$ (Figure 7a), for the exponential observation before $t = 1{,}000$ (Figure 7c). After that, the RMSE from the LETKF is almost indistinguishable from the noDA ensemble for the absolute value operator (Figure 7a), and becomes even worse for the other two observational operators

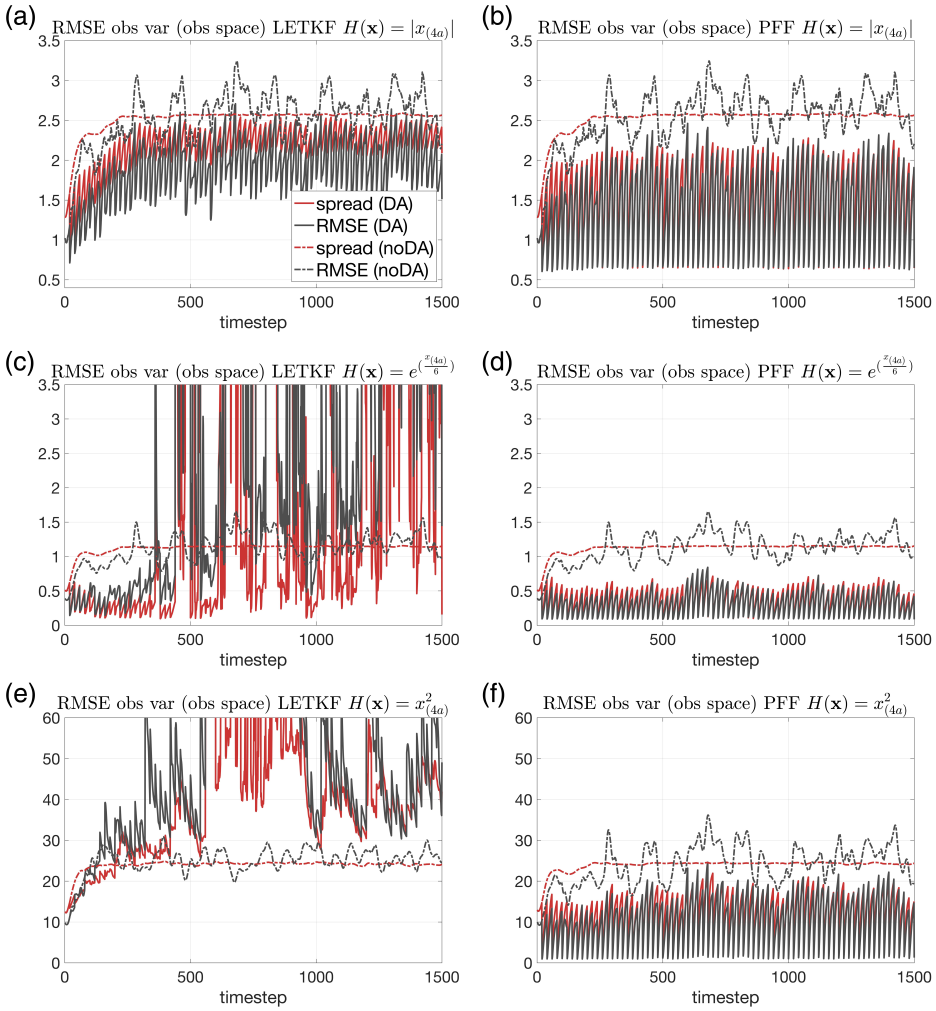

FIGURE 6 RMSE and the ensemble spread of the observed variables defined in the observational space for the nonlinear observations. The dashed lines are from the noDA ensemble, and the solid lines are from the (a, c, e) LETKF and (b, d, f) PFF. The observation operator is (a, b) the absolute value operator, (c, d) the exponential operator, and (e, f) the square operator [Colour figure can be viewed at wileyonlinelibrary.com]

(Figure 7c,e). In contrast, the RMSE from the PFF shows an improvement for all the observational operators over the noDA ensemble (Figure 7b,d,f). It is worth noting that for the PFF, the RMSE of the observed variables in model space for the absolute value (Figure 7b) and square observations (Figure 7f) at observation times are quite variable, while their values are very stable when evaluated in the observation space (Figure 6b,f). This demonstrates the problem of using the ensemble mean as a measure of the performance when the posterior is possibly not unimodal in the model space.

The rank histogram in the observational space is also used to verify the reliability of the ensemble for the nonlinear observations. Figure 8a,b shows that, for the absolute value operator, both the LETKF and the PFF have a rank histogram close to the uniform distribution. Note that the LETKF is not able to generate a bimodal posterior pdf; it can only "choose" one mode given the observation. The magnitude of the linearized observational operator is independent of the model state for the absolute value operator, meaning that this observation operator is equivalent to the linear operator when the model state is far from 0. Therefore, the incapability of generating the two modes of the posterior is a major error source for the LETKF when the observation is the absolute value operator. However, this error becomes less apparent when we transform the variables from the model state to the observational space. This can partially explain why the rank histogram of the LETKF in the observational space is close to the uniform distribution for the absolute value operator. However, for the other nonlinear observations, the rank histograms from the LETKF become U-shaped (Figure 8c,e). This is a result of biased mean, as is evident in Figure 7c,e. This suggests that the dependence of the linearized observational operator on the state is a major reason for the biased mean in the observational space for the LETKF. For the PFF, Figure 8b,d,f shows that the rank histograms for all the nonlinear operators are close to the uniform distribution, although slightly overdispersive for the exponential operator.

To demonstrate the ability of PFF to generate multimodal posterior, we compare the time series of one of the observed variables $x_{(108)}$ with square observations during $t = 250$–500 in Figure 9. Note that there can be two

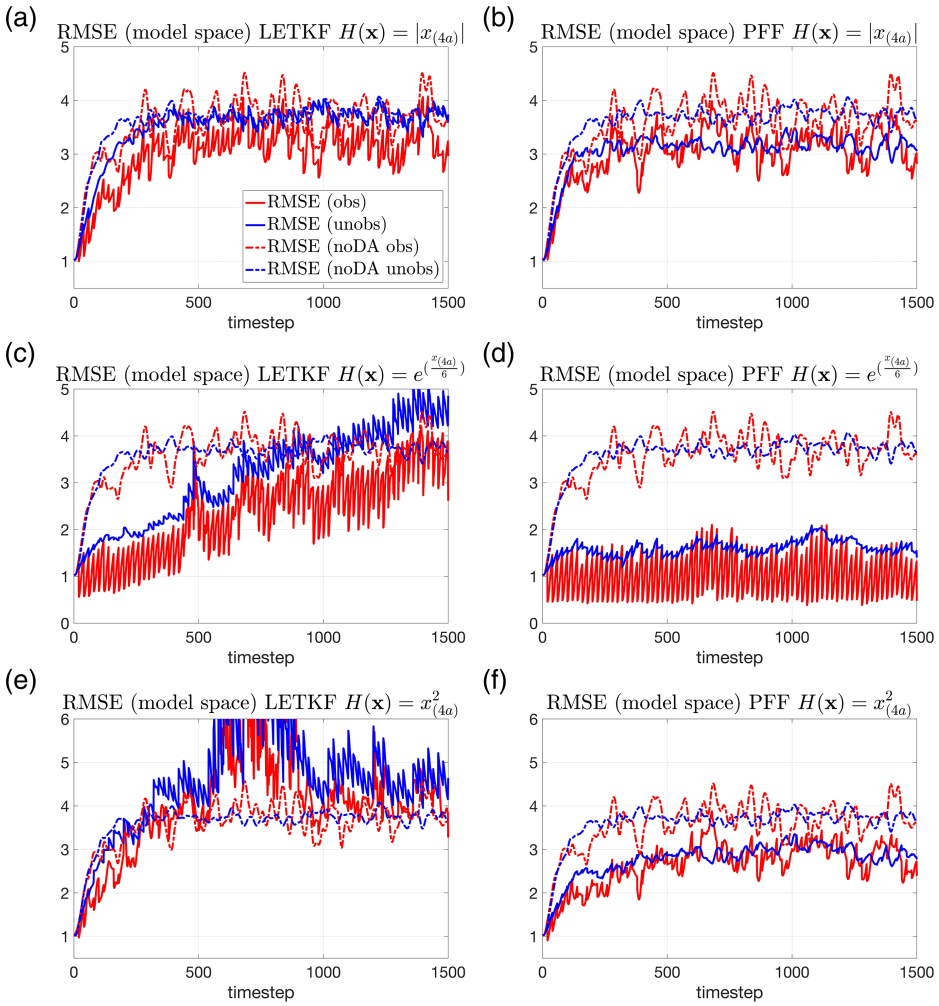

FIGURE 7 The same as Figure 3, but for the nonlinear observations. The solid lines are from the (a, c, e) LETKF and (b, d, f) PFF. The observation operator is (a, b) the absolute value operator, (c, d) the exponential operator, and (e, f) the square operator [Colour figure can be viewed at wileyonlinelibrary.com]

solutions, with the same magnitude but opposite in sign in the state space, given a square observation. We will call the solution with the opposite sign of the truth the second solution (blue dots in Figure 9). We note that, in a bimodal situation like this, the correct posterior solution is an ensemble that covers both the true and the second solutions, while individual particles follow one or the other. Both methods demonstrate an improvement of the variable evolution compared with the noDA ensemble (Figure 9). For LETKF, the whole ensemble sometimes follows the correct solutions (red dots), for example, at $t = 340$, 360, 380, 440, and 460, while the whole ensemble follows the second solution (blue dots) as well, for example, at $t = 300$, 320, 420, and 500 (Figure 9b). In contrast, for the PFF, the ensemble is able to follow both the true and the second solution at the same time (Figure 9c).

Note that when the whole ensemble from the LETKF chooses the wrong mode (i.e., follows the second solution) for the observed variables, the behavior of the unobserved variables will also be impacted by the poor covariance structures. Figure 10 shows the time series of the unobserved variable $x_{(109)}$, whose update during the data assimilation is immediately affected by its neighbor $x_{(108)}$. It is shown that after $t = 300$, the ensemble from the LETKF gradually loses track of the truth, and the spread becomes smaller (Figure 10b). Compared with the LETKF, the ensemble from the PFF has a larger spread, but follows the truth much better (Figure 10c). Although the overall spread of the unobserved variable for the PFF is large, it does not mean that the PFF loses the skill during the data assimilation. The evolution of each ensemble member still follows the truth better and in a more consistent way than the noDA ensemble (this can also be inferred from Figure 7f), suggesting that the ability of the PFF to capture the multimodal distribution of the posterior can also improve the update of the unobserved variables in its neighborhood.

It is worthwhile to mention that, in the square observation case (and other multimodal likelihood or posterior), the ensemble mean in the model state space may not be representative. For example, at $t = 420$ for the PFF (Figure 9c), almost half of the ensemble for the observed variable chooses the true solution, while another half chooses the second solution, leading to the ensemble mean

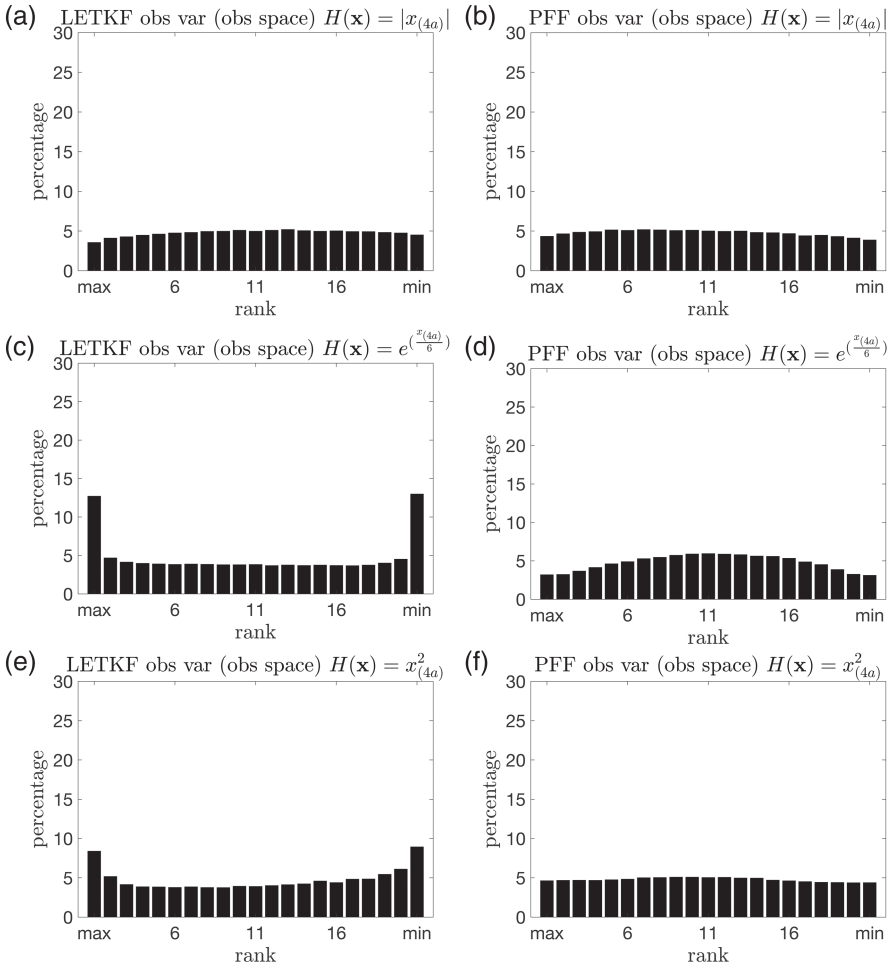

FIGURE 8 The same as Figure 5, but for the observed variables in the observational space for the nonlinear observations. The ensemble is from the (a, c, e) LETKF and (b, d, f) PFF. The observation operator is (a, b) the absolute value operator, (c, d) the exponential operator, and (e, f) the square operator

being close to the average of the two solutions, where the true posterior pdf will have little probability mass. In contrast, the behavior of the observed variable in each ensemble member from the LETKF is very similar (Figure 9b). In this case, the ensemble mean in the model state space is considered to be representative. To summarize, given the capability of PFF to generate the ensemble with multimodal distribution, we should be more cautious when using the ensemble mean in the model state space as a tool to diagnose or analyze the behavior of the ensemble. As a final note, while the posterior pdf is bimodal in observed variables, it has many more modes when considering all components together. In this sense, the behavior of the PFF is quite remarkable.

5 | THE SENSITIVITY EXPERIMENTS

In this section, we conduct several preliminary sensitivity experiments examining the effect of different settings in the PFF, including the kernel width, the number of iterations, and the prior assumption on the performance of the PFF. The preliminary aspect is related to the fact that not much research has been conducted in this area, and the research areas are too vast to cover in this paper. The same experiment setups for the sequential data assimilation experiments as described in Section 3 are used.

5.1 | The kernel width

We choose the kernel width α as the reciprocal of the number of the particles. The reason is twofold. First, we do not want α to be too large, as a large α means a strong smoothing of the particle flow through the weighting of the gradients with the kernel, in which case we lose the ability to describe the fine-scale structures in the posterior pdf. On the other hand, we do not want α to be too small either, especially with only a limited number of particles. This is because a small α means the repelling force will be significant only when particles become very close. When we only have a limited number of particles, the particles are sparse in the space. Therefore, using a small α with a limited number of particles may cause the particles to almost collapse to the mode (i.e., become too close to the mode).

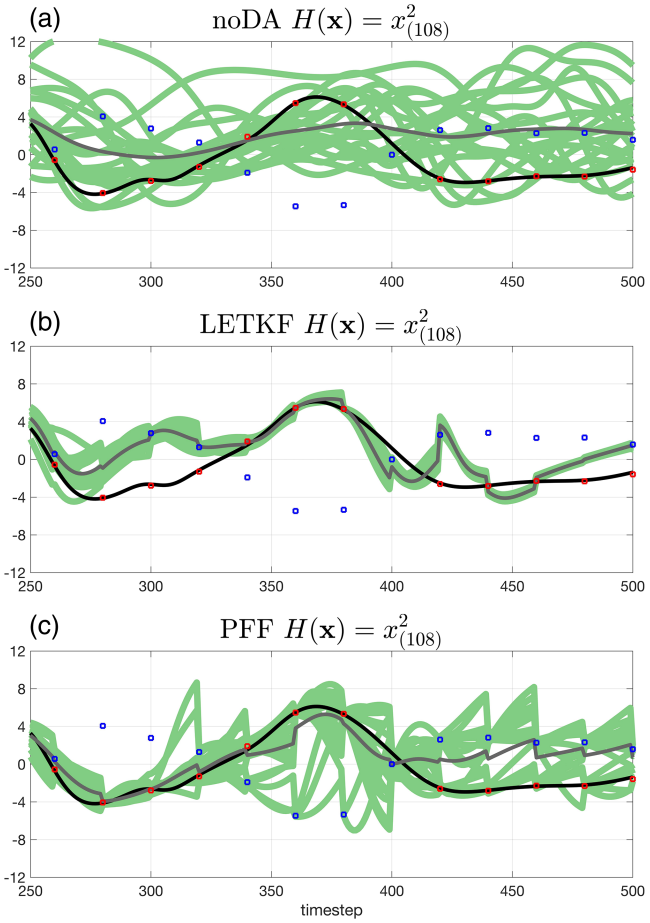

FIGURE 9 The time series of the observed variable $x_{(108)}$ during $t = 250$–500. The black line is the truth, the green lines are the ensemble members, and the gray line is the ensemble mean. The square observation is assimilated into the system. The red dots are the solution with the same sign of the truth corresponding to the observations, and the blue dots are the second solution with the opposite sign of the truth corresponding to the observations. The ensemble is from (a) noDA, (b) LETKF, and (c) PFF [Colour figure can be viewed at wileyonlinelibrary.com]

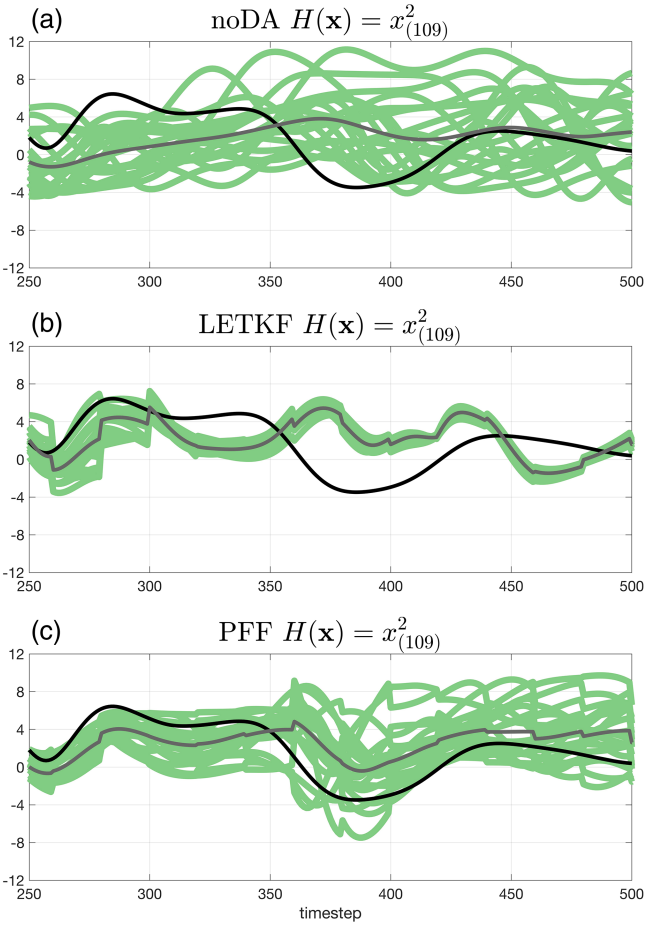

FIGURE 10 The same as Figure 9 but for the unobserved variable $x_{(109)}$. The ensemble is from (a) noDA, (b) LETKF, and (c) PFF [Colour figure can be viewed at wileyonlinelibrary.com]

Due to the above reasons, when we have more particles, we would like to make α smaller, but not too small, to keep the fine-scale structures of the particle flow at the location where each particle lies. In other words, we would expect α to be inversely proportional to the number of particles.

Indeed, the kernel width α is a tuning parameter, and there is no certain reason for why we choose it to be exactly the reciprocal of the number of particles. Therefore, the sensitivity of the performance of the PFF to α is examined in the following. Figure 11 demonstrates the effect of kernel width on the divergence of the kernel (i.e., the repelling force) for the Gaussian kernel we used in this study. Take a wider kernel (solid line in Figure 11), for example, although the range of influence for a particle is larger, the repelling force is generally smaller. This suggests that the particles can feel the repelling force from each other when their distance is large, which might lead to a larger spread of the posterior. However, the magnitude of the divergence of a wider kernel can also be too small to balance the gradient of logarithm of posterior, leading to a narrower posterior. In addition, we note that a larger kernel width means a stronger smoothing of the particle flow, which complicates the interpretation of the effect of the kernel width on the PFF.

A set of sensitivity experiments are conducted to see what the effect of the kernel width is on the PFF, using linear observations and with all other settings the same as in Section 4 except for the kernel width. Figure 12 shows the posterior marginal distribution of the variable $x_{(19)}$ (unobserved component) and $x_{(20)}$ (observed component) after the first data assimilation update for a kernel with large kernel width and another with small kernel width. A larger kernel width results in a larger spread of the posterior, through the complicated interaction between particles (Figure 12b). Figure 13 shows the total RMSE of the ensemble for 500 time steps. It is found that, when

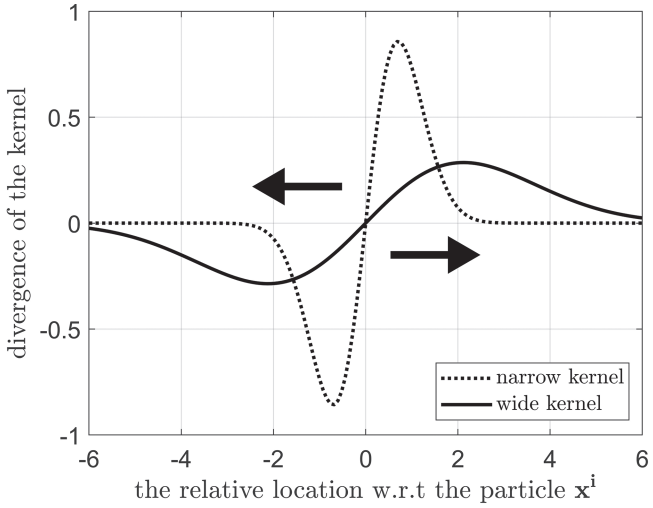

FIGURE 11 Demonstration of the effect of kernel width on the divergence of the kernel (repelling force). The system is assumed to be one-dimensional. The x axis is the relative location with respect to one of the particles, and the y axis is the divergence of the kernel that is applied on the state whose particle flow is evaluated. The dashed (solid) line represents a small (large) kernel width. The arrow represents the direction of the repelling force

the kernel width α is within the range [0.01, 0.1], the performance is comparable. When α is either too small or too large, the performance of PFF in terms of RMSE becomes suboptimal. In particular, when $\alpha = 0.001$, the RMSE becomes larger than the noDA ensemble. Figure 14 shows the rank histogram of the prior for the observed variables at the observation times. For the smaller kernel width (Figure 14a,b), the rank histogram suggests the ensemble may be either biased against the truth or underdispersive. For larger kernel width (Figure 14c,d), the rank histogram suggests that the ensemble is slightly overdispersive. This suggests that a wider (narrower) kernel leads to a larger (smaller) spread of the posterior. The good news is that a range of a factor 10 for α still leads to good performance of the filter, so the PFF is not too sensitive to this parameter, at least for the experimental settings explored here.

5.2 | The number of iterations

As in all iterative methods, we have to determine the number of iterations needed for successful interpretation of the PFF. The number of iterations is set to 500 for all previous experiments. We conduct two sets of sensitivity experiments, with one using linear observations and the other using exponential observations. Figure 15 illustrates this sensitivity by showing the total RMSE of the ensemble over 500 time steps. It is found that, once the number of iterations is over 50, the RMSE is quite similar for the linear observation case (Figure 15a). On the other hand, the RMSE becomes quasi-steady only when the number of iterations is more than 200 for the exponential observation (Figure 15b). This result suggests that PFF requires more iterations for more complicated observation operators to converge to the steady-state solution. Examinations of the iterations for PFF show that the number of iterations required can be different at different observation times (not shown). In addition, the error from the insufficient convergence at an earlier time can accumulate and affect the prior covariance at subsequent observation times. In other words, the differences between using different number of iterations are expected to be more and more pronounced if we further extend the integration time. On the other hand, we can expect fewer iterations if the observation frequency is higher. Therefore, a wise way of choosing the number of iterations for different observation operators at different observation times can improve the efficiency of the PFF. Note that the PFF is based on a steepest descent algorithm for the KL divergence. There is little experience with minimization methods for pdfs, but it is highly likely that more efficient schemes are possible. We have to leave research in this direction to a future paper.

5.3 | The prior assumption

One advantage of the PFF is that there is no assumption on the distribution of the prior, as long as we are able to derive the gradient of its logarithm analytically. In the previous section, we have assumed the prior to be Gaussian, to compare with the LETKF. However, the Gaussian assumption for the prior is definitely not optimal, for example, for hydrometeor-related variables in atmospheric models (Posselt *et al.*, 2014), or as a results of nonlinear observation operators at a previous assimilation step.

An example of a more desirable choice of the prior is a Gaussian mixture pdf. One way to construct the Gaussian mixture pdf is to first apply clustering analysis for the particles and then construct a Gaussian pdf for each cluster independently; see for example, Bengtsson *et al.* (2003). We then sum all the Gaussian pdfs with their weights proportional to the number of particles in each cluster to form the Gaussian mixture pdf. As for the way to conduct the clustering analysis, we should note that any method requiring repeated calculations of the distance between particles can be computationally prohibitive for a high-dimensional system. For example, k-means clustering can be too expensive. Other methods, such as agglomerative hierarchical clustering, in which we only need to calculate the distance between particles once, might be affordable. Although the agglomerative hierarchical clustering method seems

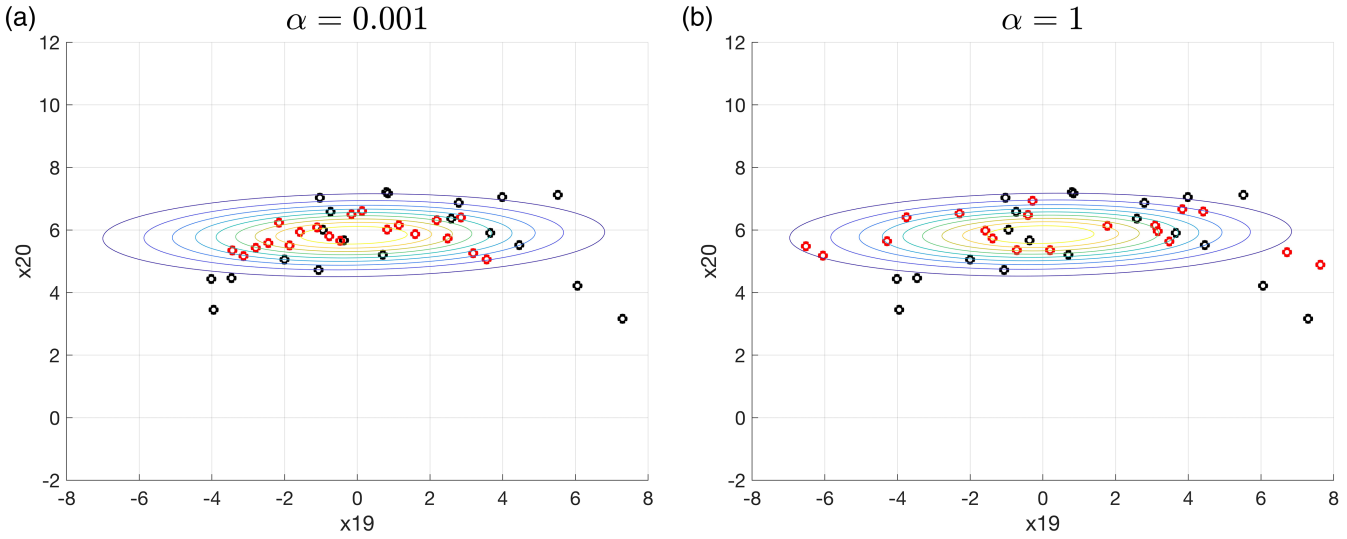

FIGURE 12 The same as Figure 3, but the kernels used here are both matrix-valued kernels yet with different kernel widths. (a) A narrow kernel width $\alpha = 0.001$. (b) A wide kernel width $\alpha = 1$ [Colour figure can be viewed at wileyonlinelibrary.com]

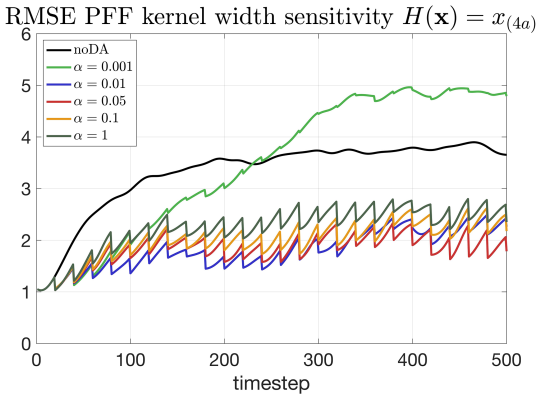

FIGURE 13 The total RMSE of the ensemble from the PFF, assimilating linear observations, with different kernel widths [Colour figure can be viewed at wileyonlinelibrary.com]

promising, we found that it is difficult to construct the Gaussian mixture prior in the system with the square observations. The reason is in the following: despite many of the marginal distributions for the observed variables in the prior being bimodal distributed, in which case two clusters might seem to be enough, the joint distribution for all the observed variables can have many more modes. For example, we can have four modes for a two-dimensional system if the two variables are independent and both have bimodal marginal distribution. Given that we only have 20 particles, it is difficult to construct a proper Gaussian mixture prior based on the clustering method: the number of modes is just too high.

Another possible way is to assume the prior to be a Gaussian mixture with equal weights, and with the same covariance $\mathbf{D}$ for each component. The mean of each component is set to be the state of each particle $\mathbf{x}_0^i$, and the Gaussian mixture can be written as

$$p(\mathbf{x}_0) = \frac{1}{N_p}\sum_{i=1}^{N_p} N(\mathbf{x}_0^i, \mathbf{D}) \tag{40}$$

In this case, the problem of determining the prior reduces to determining the covariance in each component of the Gaussian mixture. We can determine this covariance $\mathbf{D}$ by assuming that the covariance of the Gaussian mixture is equal to the sample covariance of all particles together. The covariance of the Gaussian mixture can be written as

$$\begin{aligned} \mathrm{Cov}(\mathbf{x}_0) &= \mathbf{D} + \frac{1}{N_p}\sum_{i=1}^{N_p}(\mathbf{x}_0^i - \overline{\mathbf{x}_0})(\mathbf{x}_0^i - \overline{\mathbf{x}_0})^T \\ \overline{\mathbf{x}_0} &= \frac{1}{N_p}\sum_{i=1}^{N_p}\mathbf{x}_0^i \end{aligned} \tag{41}$$

where $\overline{\mathbf{x}_0}$ is the ensemble mean of the prior. The sample covariance of the particles is

$$\mathbf{B} = Cov(\{\mathbf{x}_0^i\}_{i=1}^{N_p}) = \frac{1}{N_p - 1}\sum_{i=1}^{N_p}(\mathbf{x}_0^i - \overline{\mathbf{x}_0})(\mathbf{x}_0^i - \overline{\mathbf{x}_0})^T \tag{42}$$

We can immediately obtain the covariance matrix $\mathbf{D}$ as

$$\mathbf{D} = \frac{\mathbf{B}}{N_p} \tag{43}$$

Experiments with the Gaussian mixture in Equation (40) show that the covariance for each component $\mathbf{D}$ is too narrow, making the update of each particle very limited (not shown). We can improve this update by

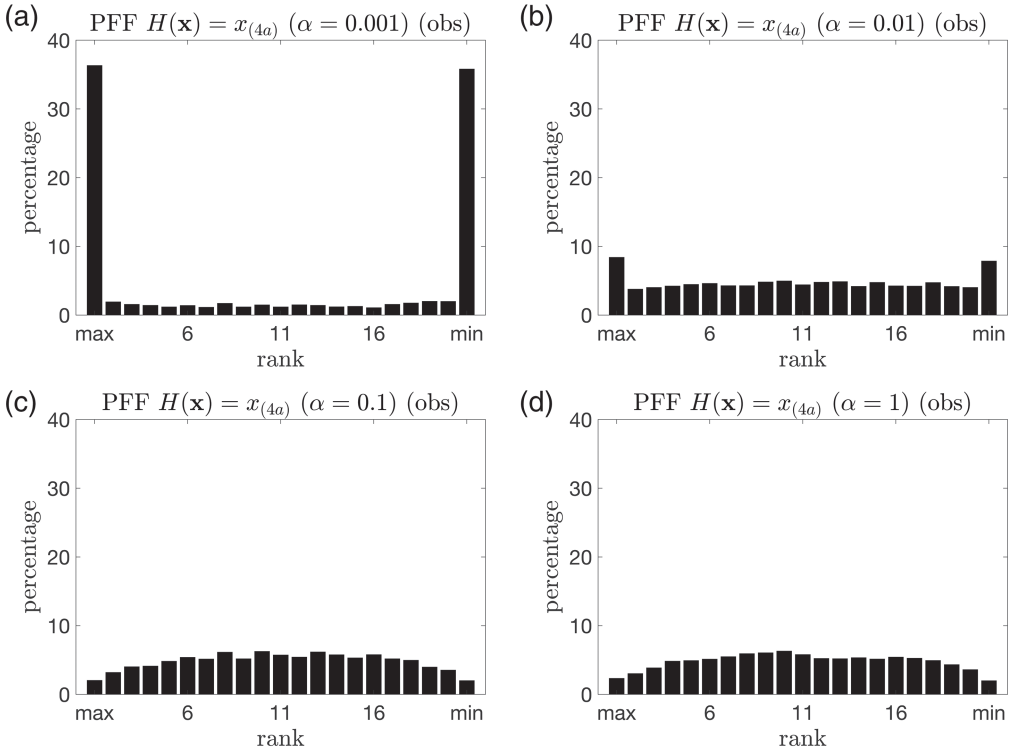

FIGURE 14 The same as Figure 5, but for the PFF with different kernel width. (a) α = 0.001, (b) α = 0.01, (c) α = 0.1, and (d) α = 1. For the experiment with α = 0.05, see Figure 5c

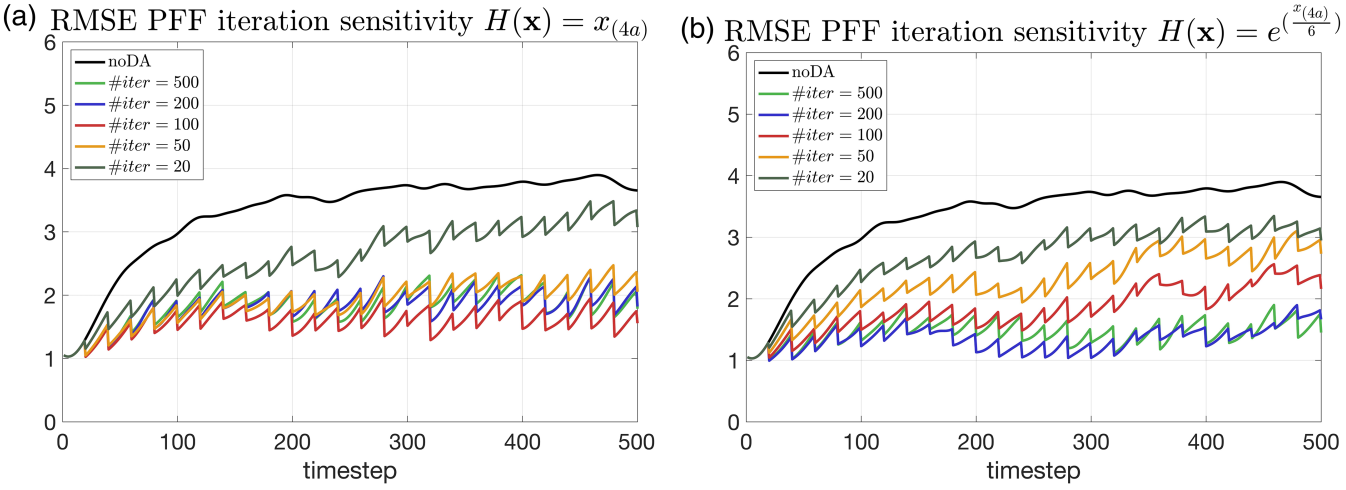

FIGURE 15 The same as Figure 13, but for a different number of iterations for the PFF using (a) the linear observation and (b) the exponential observation [Colour figure can be viewed at wileyonlinelibrary.com]

making **D** wider through multiplication with a scalar, so that each component is able to interact with each other. However, we note that making **D** wider is to some extent equivalent to inflating the prior. We should not make **D** so wide that the information of the prior will be lost. Nevertheless, we can still tune the width of **D** to seek an optimal width for our system. We have tried this Gaussian mixture prior in the system with square observations, but the RMSE of this ensemble does not show significant improvement over the ensemble with Gaussian prior (not shown). This might be due to the characteristics of the current system, since the relationship even between neighboring variables is generally weak in the Lorenz 1996 model. We still expect an improvement of PFF performance using the Gaussian mixture prior in a real atmospheric model over the Gaussian prior, since some nonlinear relationships, such as hydrometeor variables, will be apparent. Detailed examinations of different prior assumptions on the performance of the PFF are needed in future studies.

6 | CONCLUSIONS AND DISCUSSION

The particle flow filter (PFF) is a recently developed Monte Carlo filter based on a deterministic flow, which naturally avoids the weight degeneracy problem and thus has potential to be applied to high-dimensional problems. The PFF optimally transforms the particles from the prior distribution to the posterior distribution, maintaining equal weight for all the particles at all iteration steps. With the assumption that the particle flow is embedded in a reproducing kernel Hilbert space (RKHS), we are able to derive an analytical expression for the particle flow that minimizes the Kullback–Leibler divergence (KL divergence) between intermediate pdfs and the posterior pdf, starting at the prior.

The particle flow is composed of two terms: the weighted average of the gradient of the logarithm of the posterior and the divergence of the kernel. With the former term alone, the particles are driven to the mode of the posterior. This can be demonstrated when we have only one particle. In this case, the divergence of the kernel vanishes and the particle flow is equivalent to a 3DVar. The divergence of the kernel acts to repel the particles away from each other. When the summation of two terms for all the particles balance each other, the particle distribution will describe the posterior distribution.

In the limit of infinite number of particles, the solution from the PFF is independent of the choice of the kernel. However, since we have only a finite number of particles, the choice of the kernel becomes critical. When the PFF was first developed, a scalar Gaussian kernel was found to work well in a relatively low-dimensional system with dense observations. However, we find that in the sparsely observed high-dimensional system, in which the variance of the gradient of posterior among variables can be very large, the scalar kernel fails to maintain the variance of the marginal distribution of the observed variables. To tackle this problem, a new matrix-valued Gaussian kernel is proposed in this study. The proposed kernel is a diagonal matrix with a different Gaussian scalar kernel in the diagonal entries. The advantage of this kernel is that it independently measures the distance between particles in each direction, ensuring that the marginal distributions will not collapse.

The PFF with the newly proposed matrix-valued kernel is tested with a sequential data assimilation experiment in a 1,000-dimensional Lorenz 96 system, with only 25% of the system observed every 20 time steps. With linear observations, the performance of the PFF is similar to a well-tuned LETKF. Note that, with a proper choice of kernel width, the PFF does not require inflation of the prior, while an inflated prior is needed for the LETKF to achieve similar results. With nonlinear observations, the PFF outperforms the LETKF in terms of the RMSE in observational space and the rank histogram of the prior at observation times. We have separately examined two aspects of the nonlinear observations: a multimodal structure in the likelihood and the dependency of the linearized observation operator on the state. Since the PFF is able to evaluate the linearized observation operator locally for each particle, the PFF can capture the multimodal distribution of the likelihood in the absolute value and square observation operators, which also improves the update of the unobserved variables through a better covariance structure in the system. On the other hand, since the PFF iteratively updates the linearized observation operator, it can much better capture the nonlinear relation between model state and observations.

The sensitivity of settings in the PFF to its performance is also examined, but much more work is needed. In terms of kernel width, it is found that a wide (narrow) kernel tends to make the posterior wider (narrower), but a strong point of the methodology is that the sensitivity to the scaling factor is small with good performance over a range of a factor 10. An optimal width is found when the scaling factor is the reciprocal of the number of particles. While we explored a diagonal matrix-valued kernel, a possible extension is to explore the off-diagonal elements to provide a smoother repelling force. We did not need that in our experiments, but this might be useful in more realistic models. In terms of the iteration number, it is found that fewer iterations are needed for the linear observation than the nonlinear observations to reach a steady-state posterior solution. In addition, the iterations needed at different observation times can be quite different. We use the quasi-Newton method to accelerate the convergence, with the preconditioner chosen to be the localized background error covariance from the particles, but other methods might be more beneficial. In terms of the assumption for the prior, we propose different ways of efficiently constructing a Gaussian mixture prior. The sensitivity experiments show that the PFF with a Gaussian mixture prior does not show significant improvement over the PFF with a Gaussian prior, mainly related to the difficulty of applying an efficient clustering algorithm to define the mixtures. Furthermore, we pushed the method by only using 20 particles, in which case clustering is perhaps overdoing it. Other possibilities include a hybrid covariance between ensemble covariance and a static (climatology) covariance. This might especially be of interest when Gaussian mixtures are used with a small ensemble size.

When the PFF is applied to a real atmospheric model with complex observation operators, the adjoint of the

observational operator may not always be available. In the ensemble Kalman filter, the ensemble covariance between state and observation space can be used to replace the adjoint model. However, when the observation is highly nonlinear, the ensemble covariance will not be accurate. An alternative solution to avoid the adjoint is to assume that the observational operator is embedded in a RKHS (Pulido *et al.*, 2019). A similar trick is used as in the PFF, and the gradient of the observational operator can be obtained without need for the full adjoint model. Pulido *et al.*, 2019 have also shown that the performance of PFF using the RKHS approximation for the linearized observation operator is better than using the ensemble covariance approximation.

Finally, the PFF that has been developed here is a filter, but it can easily be extended to a smoother. Since the PFF is computationally similar to an ensemble of 3DVars, this would be similar to an extension to an ensemble 4DVars. Indeed, as an example, the "Ensemble of data assimilations" promoted by ECMWF, which is essentially an ensemble smoother, can be transformed relatively easy into a fully nonlinear data-assimilation system via the PFF. The main difference is that the 4DVars will have to communicate at every iteration by sending over full state vectors from one minimization to the other. One can also envisage a scheme in which communication is not at every iteration step to speed up calculations. There remains much to do before this is reality, but there is a clear path ahead.

ACKNOWLEDGMENTS

The authors thank the Cooperative Institute for Research of the Atmosphere, and PjvL thanks the European Research Council for funding via the CUNDA project under number 694509.

ORCID

Chih-Chi Hu https://orcid.org/0000-0002-3020-8975
Peter Jan van Leeuwen https://orcid.org/0000-0003-2325-5340

REFERENCES

Anderson, J.L. (2007) An adaptive covariance inflation error correction algorithm for ensemble filters. *Tellus A*, 59, 210–224. https://doi.org/10.1111/j.1600-0870.2006.00216.x.

Bengtsson, T., Snyder, C. and Nychka, D. (2003) Toward a nonlinear ensemble filter for high-dimensional systems. *Journal of Geophysical Research*, 108, 8775. https://doi.org/10.1029/2002JD002900.

Burgers, G., van Leeuwen, P.J. and Evensen, G. (1998) Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review*, 126(6), 1719–1724.

Chorin, A.J., Morzfeld, M. and Tu, X. (2010) Interpolation and iteration for nonlinear filters. *Communications in Applied Mathematics and Computational Science*, 5, 221–240.

Daum, F. and Huang, J. (2011) Particle flow for nonlinear filters. In *Proceedings of IEEE International Conference of Acoustics, Speech, and Signal Processing*, ICASSP 2011, Prague, 22–27 May 2011, pp 5920–5923, https://doi.org/10.1109/ICASSP.2011.5947709.

Evensen, G. (1994) Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte-Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99, 10143–10162.

Hamill, T.M. (2001) Interpretation of rank histograms for verifying ensemble forecasts. *Monthly Weather Review*, 129, 550–560. https://doi.org/10.1175/15200493(2001)129<0550:IORHFV>2.0.CO;2.

Houtekamer, P.L. and Mitchell, H.L. (1998) Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Review*, 126, 796–811.

Jones, T.A., Knopfmeier, K., Wheatley, D., Creager, G., Minnis, P. and Palikonda, R. (2016) Storm-scale data assimilation and ensemble forecasting with the NSSL experimental warn-on-forecast system. Part II: combined radar and satellite data experiments. *Weather and Forecasting*, 31, 297–327. https://doi.org/10.1175/WAF-D-15-0107.1.

Liu, Q. and Wang, D. (2016) Stein variational gradient descent: a general purpose Bayesian inference algorithm. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16, Barcelona, Spain, 5–10 December 2016*, pp 2378–2386, arXiv: 1608.04471.

Lu, J., Lu, Y. and Nolen, J. (2019) Scaling limit of the Stein variational gradient descent: the mean field regime. *SIAM Journal on Mathematical Analysis*, 51, 648–671. https://doi.org/10.1137/18M1187611.

Minamide, M. and Zhang, F. (2019) An adaptive background error inflation method for assimilating all-sky radiances. *Quarterly Journal of the Royal Meteorological Society*, 145, 805–823. https://doi.org/10.1002/qj.3466.

Morzfeld, M., Tu, X., Atkins, E. and Chorin, A.J. (2012) A random map implementation of implicit filters. *Journal of Computational Physics*, 231, 2049–2066.

Nocedal, J. and Wright, S.J. (2006) *Numerical Optimization: Springer Series in Operations Research and Financial Engineering*. New York: Springer.

Posselt, D.J., Hodyss, D. and Bishop, C.H. (2014) Errors in ensemble Kalman smoother estimates of cloud microphysical parameters. *Monthly Weather Review*, 142, 1631–1654. https://doi.org/10.1175/MWR-D-13-00290.1.

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

Poterjoy, J., Wicker, L. and Buehner, M. (2019) Progress toward the application of a localized particle filter for numerical weather prediction. *Monthly Weather Review*, 147, 1107–1126. https://doi.org/10.1175/MWR-D-17-0344.1.

Pulido, M. and van Leeuwen, P.J. (2019) Sequential Monte Carlo with kernel embedded mappings: the mapping particle filter. *Journal of Computational Physics*, 396, 400–415. https://doi.org/10.1016/j.jcp.2019.06.060.

Pulido, M., vanLeeuwen, P.J. and Posselt, D.J. (2019) Kernel embedded nonlinear observational mappings in the variational mapping particle filter. In: Rodrigues, J., Cardoso, P.J.S., Monteiro, J., Lam, R., Krzhizhanovskaya, V.V., Lees, M.H., Dongarra, J.J. and Sloot, P.M.A. (Eds.) *Computational Science – ICCS 2019. ICCS 2019*. Lecture Notes in Computer Science, Vol. 11539. Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-030-22747-0_11.

Snyder, C. and Zhang, F. (2003) Assimilation of simulated Doppler radar observations with an ensemble Kalman filter. *Monthly Weather Review*, 131, 1663–1677. https://doi.org/10.1175//2555.1.

Snyder, C., Bengtsson, T., Bickel, P. and Anderson, J. (2008) Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136, 4629–4640. https://doi.org/10.1175/2008MWR2529.1.

Van Leeuwen, P. J. (2003) Nonlinear ensemble data assimilation for the ocean. In *Seminar Recent Developments in Data Assimilation for Atmosphere and Ocean, 8–12 September 2003*. ECMWF, Reading, UK.

Van Leeuwen, P.J. (2009) Particle filtering in geophysical systems. *Monthly Weather Review*, 137, 4089–4114. https://doi.org/10.1175/2009MWR2835.1.

Van Leeuwen, P.J. (2010) Nonlinear data assimilation in geosciences: an extremely efficient particle filter. *Quarterly Journal of the Royal Meteorological Society*, 136, 1991–1999. https://doi.org/10.1002/qj.699.

Van Leeuwen, P.J. (2020) A consistent interpretation of the stochastic version of the Ensemble Kalman Filter. *Quarterly Journal of the Royal Meteorological Society*, 146, 2815–2825. https://doi.org/10.1002/qj.3819.

Van Leeuwen, P.J., Künsch, H.R., Nerger, L., Potthast, R. and Reich, S. (2019) Particle filters for high-dimensional geoscience applications: a review. *Quarterly Journal of the Royal Meteorological Society*, 145, 2335–2365. https://doi.org/10.1002/qj.3551.

Whitaker, J.S. and Hamill, T.M. (2012) Evaluating methods to account for system errors in ensemble data assimilation. *Monthly Weather Review*, 140, 3078–3089. https://doi.org/10.1175/MWR-D-11-00276.1.

Ying, Y. and Zhang, F. (2015) An adaptive covariance relaxation method for ensemble data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 141, 2898–2906. https://doi.org/10.1002/qj.2576.

Zhang, F., Snyder, C. and Sun, J. (2004) Impacts of initial estimate and observation availability on convective-scale data assimilation with an ensemble Kalman filter. *Monthly Weather Review*, 132, 1238–1253. https://doi.org/10.1175/1520-0493(2004)132<1238:IOIEAO>2.0.CO;2.

Zhang, F., Weng, Y., Sippel, J.A., Meng, Z. and Bishop, C.H. (2009) Cloud-resolving hurricane initialization and prediction through assimilation of Doppler radar observations with an Ensemble Kalman Filter. *Monthly Weather Review*, 137, 2105–2125. https://doi.org/10.1175/2009MWR2645.1.

How to cite this article: Hu C-C, van Leeuwen PJ. A particle flow filter for high-dimensional system applications. *Q J R Meteorol Soc*. 2021;147:2352–2374. https://doi.org/10.1002/qj.4028

APPENDIX A. THE DERIVATION OF THE PFF IN EQUATIONS (6) AND (7)

The Kullback–Leibler divergence (KL divergence) between the intermediate pdf at pseudo time s q_s and targeted pdf $p(\mathbf{x}|\mathbf{y})$ is given by Equation (A1):

$$KL(q_s) := \int q_s(\mathbf{x}) \log\left(\frac{q_s(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})}\right) d\mathbf{x} \tag{A1}$$

The goal of this derivation is to find the appropriate flow field $\mathbf{f}_s$, such that the KL divergence decreases with pseudo time:

$$\frac{d}{ds}\mathbf{x} = \mathbf{f}_s(\mathbf{x}), \mathrm{s} \in [0, \infty] \tag{A2}$$

$$KL(q_{s+\Delta s}) \leq KL(q_s) \tag{A3}$$

where $q_{s+\Delta s}$ is the imtermediate pdf at pseudo time $s + \Delta s$.

To achieve the goal, we require that the derivate of KL divergence to pseudo time is negative:

$$\frac{dKL}{ds} \leq 0 \tag{A4}$$

That is,

$$\begin{aligned}\frac{dKL}{ds} &= \frac{d}{ds}\int q_s(\mathbf{x}) \log\left(\frac{q_s(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})}\right) d\mathbf{x} \\ &= \int \frac{\partial}{\partial s}\left\{q_s(\mathbf{x}) \log\left(\frac{q_s(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})}\right)\right\} d\mathbf{x} \\ &= \int \frac{\partial q_s(\mathbf{x})}{\partial s}\left\{\log\left(\frac{q_s(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})}\right) + 1\right\} d\mathbf{x} \leq 0\end{aligned} \tag{A5}$$

Since we assume there is only advection in the transformation of the state $\mathbf{x}$, the pseudo time rate change of the intermediate pdf q_s is characterized by the corresponding Liouville equation:

$$\frac{\partial q_s(\mathbf{x})}{\partial s} + \nabla_{\mathbf{x}} \cdot (\mathbf{f}_s(\mathbf{x}) q_s(\mathbf{x})) = 0 \tag{A6}$$

Based on the Liouville equation (Equation (A6)), Equation (A5) can be written as

$$\frac{dKL}{ds} = -\int \nabla_{\mathbf{x}} \cdot (\mathbf{f}_s(\mathbf{x}) q_s(\mathbf{x}))\left\{\log\left(\frac{q_s(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})}\right) + 1\right\} d\mathbf{x} \tag{A7}$$

Integrating by parts, Equation (A7) can be further written as

$$\frac{dKL}{ds} = \int \mathbf{f}_s(\mathbf{x}) q_s(\mathbf{x}) \cdot \nabla_{\mathbf{x}}\left\{\log\left(\frac{q_s(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})}\right) + 1\right\} d\mathbf{x} \tag{A8}$$

We have used the fact that q_s should approach zero on the boundary to assure that the integral of q_s in the whole

domain is finite. Equation (A8) can be further written as

$$\begin{aligned}\frac{dKL}{ds} &= \int \mathbf{f}_s(\mathbf{x})q_s(\mathbf{x}) \cdot \nabla_{\mathbf{x}}\{\log(q_s(\mathbf{x})) - \log(p(\mathbf{x}|\mathbf{y}))\}d\mathbf{x} \\ &= \int \mathbf{f}_s(\mathbf{x}) \cdot \nabla_{\mathbf{x}} q_s(\mathbf{x}) - \mathbf{f}_s(\mathbf{x})q_s(\mathbf{x}) \cdot \nabla_{\mathbf{x}} \log(p(\mathbf{x}|\mathbf{y}))d\mathbf{x} \\ &= -\int q_s(\mathbf{x})\nabla_{\mathbf{x}} \cdot \mathbf{f}_s(\mathbf{x}) + \mathbf{f}_s(\mathbf{x})q_s(\mathbf{x}) \cdot \nabla_{\mathbf{x}} \log(p(\mathbf{x}|\mathbf{y}))d\mathbf{x}\end{aligned} \tag{A9}$$

where the integration by parts has been used again from line two to line three in Equation (A9).

We now assume the solution of $\mathbf{f}_s$ is embedded in a reproducing kernel Hilbert space (RKHS). Every function in a RKHS can be written as

$$\mathbf{f}_s(\mathbf{x}) = \langle \mathbf{K}(\mathbf{x}, \cdot), \mathbf{f}_s(\cdot) \rangle \in \mathbb{R}^{n_x} \tag{A10}$$

where $\mathbf{K}$ is the matrix-valued kernel. The bracket is the inner product in the RKHS. Based on Equation (A10), the divergence of $\mathbf{f}_s$ can be written as

$$\nabla_{\mathbf{x}} \cdot [\mathbf{f}_s(\mathbf{x})] = \nabla_{\mathbf{x}} \cdot \langle \mathbf{K}(\mathbf{x}, \cdot), \mathbf{f}_s(\cdot) \rangle = \langle \nabla_{\mathbf{x}} \cdot \mathbf{K}(\mathbf{x}, \cdot), \mathbf{f}_s(\cdot) \rangle_1 \in \mathbb{R} \tag{A11}$$

Based on Equations (A10) and (A11), the rate of change of the KL divergence in Equation (A9) can be written as

$$\begin{aligned}&-\int q_s(\mathbf{x})\langle \nabla_{\mathbf{x}} \cdot \mathbf{K}(\mathbf{x}, \cdot), \mathbf{f}_s(\cdot) \rangle_1 + \langle \mathbf{K}(\mathbf{x}, \cdot), \mathbf{f}_s(\cdot) \rangle q_s(\mathbf{x}) \cdot \nabla_{\mathbf{x}} \\ &\quad \log(p(\mathbf{x}|\mathbf{y}))d\mathbf{x} \\ &= -\int q_s(\mathbf{x})\{\langle \nabla_{\mathbf{x}} \cdot \mathbf{K}(\mathbf{x}, \cdot), \mathbf{f}_s(\cdot) \rangle_1 + \langle \mathbf{K}(\mathbf{x}, \cdot)\nabla_{\mathbf{x}} \\ &\quad \log(p(\mathbf{x}|\mathbf{y})), \mathbf{f}_s(\cdot) \rangle_1 \}d\mathbf{x} \\ &= \left\langle -\int q_s(\mathbf{x})\{\nabla_{\mathbf{x}} \cdot \mathbf{K}(\mathbf{x}, \cdot) + \mathbf{K}(\mathbf{x}, \cdot)\nabla_{\mathbf{x}} \right. \\ &\quad \left. \log(p(\mathbf{x}|\mathbf{y}))\}d\mathbf{x}, \mathbf{f}_s(\cdot) \right\rangle_1\end{aligned} \tag{A12}$$

To make the pseudo time rate change of KL-divergence negative, that is, Equation (A4), we choose $\mathbf{f}_s$ based on Equation (A12) as

$$\begin{aligned}\mathbf{f}_s(\cdot) &= \mathbf{D}\int q_s(\mathbf{x})\{\nabla_{\mathbf{x}} \cdot \mathbf{K}(\mathbf{x}, \cdot) + \mathbf{K}(\mathbf{x}, \cdot)\nabla_{\mathbf{x}} \log(p(\mathbf{x}|\mathbf{y}))\}d\mathbf{x} \\ &= \mathbf{D}\mathbf{I}_f\end{aligned} \tag{A13}$$

in which $\mathbf{D}$ is a positive definite matrix included to ensure that $\mathbf{f}_s$ has the right physical dimension. With this choice, and abbreviating the integral as $\mathbf{I}_f$, Equation (A12) becomes

$$\frac{dKL}{ds} = \langle -\mathbf{I}_f, \mathbf{D}\mathbf{I}_f \rangle \leq 0 \tag{A14}$$

Here we used that $\mathbf{D}$ is positive definite with respect to the inner product. Note that $\mathbf{f}_s$ is pointing in the direction with the greatest descent of KL divergence, modified by matrix $\mathbf{D}$.

To implement Equation (A13), we need to represent $q_s(\mathbf{x})$ with the particle representation. Assuming the particles are $\{\mathbf{x}_s^i\}_{i=1}^{N_p}$, then

$$q_s(\mathbf{x}) = \frac{1}{N_p}\sum_{i=1}^{N_p} \delta(\mathbf{x} - \mathbf{x}_\mathbf{s}^\mathbf{i}) \tag{A15}$$

With Equations (A13) and (A15), we can obtain the solution of the particle flow:

$$\mathbf{f}_s(\cdot) = \frac{1}{N_p}\mathbf{D}\sum_{i=1}^{N_p} \nabla_{\mathbf{x}_s^i} \cdot \mathbf{K}(\mathbf{x}_s^i, \cdot) + \mathbf{K}(\mathbf{x}_s^i, \cdot)\nabla_{\mathbf{x}_s^i} \log(p(\mathbf{x}_s^i|\mathbf{y})) \tag{A16}$$

where

$$\nabla_{\mathbf{x}_s^i} \cdot \mathbf{K}(\mathbf{x}_s^i, \cdot) = \nabla_{\mathbf{x}} \cdot \mathbf{K}(\mathbf{x}, \cdot)|_{\mathbf{x}=\mathbf{x}_s^i} \tag{A17}$$

and

$$\nabla_{\mathbf{x}_s^i} \log(p(\mathbf{x}_s^i|\mathbf{y})) = \nabla_{\mathbf{x}} \log(p(\mathbf{x}|\mathbf{y}))|_{\mathbf{x}=\mathbf{x}_s^i} \tag{A18}$$