DISSERTATION

HIGH-DIMENSIONAL NONLINEAR DATA ASSIMILATION WITH NON-GAUSSIAN OBSERVATION ERRORS FOR THE GEOSCIENCES

Submitted by

Chih-Chi Hu

Department of Atmospheric Science

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2023

Doctoral Committee:

Advisor: Peter Jan van Leeuwen

Christian Kummerow
Jeffrey Anderson
Michael Bell
Michael Kirby

ABSTRACT

HIGH-DIMENSIONAL NONLINEAR DATA ASSIMILATION WITH NON-GAUSSIAN OBSERVATION ERRORS FOR THE GEOSCIENCES

Data assimilation (DA) plays an indispensable role in modern weather forecasting. DA aims to provide better initial conditions for the model by combining the model forecast and the observations. However, modern DA methods for weather forecasting rely on linear and Gaussian assumptions to seek efficient solutions. These assumptions can be invalid, e.g., for problems associated with clouds, or for the assimilation of remotely-sensed observations. Some of these observations are either discarded, or not used properly due to these inappropriate assumptions in DA. Therefore, the goal of this dissertation is to seek solutions to tackle the issues arising from the linear and Gaussian assumptions in DA.

This dissertation can be divided into two parts. In the first part, we explore the potential of the particle flow filter (PFF) in high dimensional systems. First, we tested the PFF in the 1000-dimensional Lorenz 96 model. The key innovation is we find that using a matrix kernel in the PFF can prevent the collapse of particles along the observed directions, for a sparsely observed and high-dimensional system with only a small number of particles. We also demonstrate that the PFF is able to represent a multi-modal posterior distribution in a high-dimensional space. Next, in order to apply the PFF for the atmospheric problem, we devise a parallel algorithm for PFF in the Data Assimilation Research Testbed (DART), called PFF-DART. A two-step PFF was developed that closely resembles the original PFF algorithm. A year-long cycling data assimilation experiment with a simplified atmospheric general circulation model shows PFF-DART is able to produce stable and comparable results to the Ensemble Adjustment Kalman Filter (EAKF) for linear and Gaussian observations. Moreover, PFF-DART can better assimilate the non-linear observations and reduce the errors of the ensemble, compared to the EAKF.

In the second part, we shift our focus to the observation error in data assimilation. Traditionally, observation errors have been assumed to follow a Gaussian distribution mainly for two reasons: it is difficult to estimate observation error statistics beyond its second moment, and most of the DA methods assume a Gaussian observation error by construction. We developed the so-called Deconvolution-based Observation Error Estimation (DOEE), that can estimate the full distribution of the observation error. We apply DOEE to the all-sky microwave radiances and show that they indeed have non-Gaussian observation errors, especially in a cloudy and humid environment. Next, in order to incorporate the non-Gaussian observation errors into variational methods, we explore an evolving-Gaussian approach, that essentially uses a state dependent Gaussian observation error in each outer loop of the minimization. We demonstrate the merits of this method in an idealized experiment, and implemented it in the Integrated Forecasting System of the European Centre for Medium-Range Weather Forecasts. Preliminary results show improvement for the short-term forecast of lower-tropospheric humidity, cloud, and precipitation when the observation error models of a small set of microwave channels are replaced by the non-Gaussian error models.

In all, this dissertation provides possible solutions for outstanding non-linear and non-Gaussian data assimilation problems in high-dimension systems. While there are still important remaining issues, we hope this dissertation lays a foundation for the future non-linear and non-Gaussian data assimilation research and practice.

ACKNOWLEDGEMENTS

I would like to first express my gratitude to my advisor, Peter Jan van Leeuwen. First, thank Peter Jan for selecting me as a PhD student in his group 4 years ago, despite my very little knowledge of data assimilation prior to joining CSU. The opportunity to learn and to work on cutting-edge data assimilation research under this guidance has been a truly remarkable experience. I enjoy every discussions with Peter Jan, especially when he explains new ideas to me with so much passion, filling the whiteboard with equations. I am especially grateful for his patience during my first year when COVID started. Despite my slow progress and repeated questions, he always explained concepts with patience. There have been many times when I felt I may have taken up a significant amount of his time seeking assistance, yet he never hesitated to provide support. Also, thank Peter Jan for giving me so many opportunities to attend conferences, present my work, and collaborate with other scientists. These experiences have been invaluable for me. Finally, thank Peter Jan for his mental support. When I found myself overwhelmed with pessimism regarding my work, he always provided consistently encouragement that helped me regain confidence. It would have been very challenging for me to navigate through all the obstacles without his support and motivation.

I sincerely appreciate Jeffrey Anderson for his great assistance in my PhD research. I have learned a lot from Jeff, ranging from learning important data assimilation research, to writing FORTRAN codes, and grasping the concept of parallel computing. Without his guidance, it would have been impossible for me to write a parallelizable code of PFF in DART by myself. I am grateful to Jeff for generously spending a significant amount of his time to meet with me and share his expertise. In addition, I have gained a lot professional and academic experiences from him. Thank Jeff for the support and guidance to my career journey.

I am thankful for my PhD committee members: Christian Kummerow, Jeffrey Anderson, Michael Bell and Michael Kirby, for their time in going over my preliminary exam, participating the committee meeting, and reviewing my dissertation. All of their feedback and comments have helped me improve my professional and science communication skills.

I would also like to convey my gratitude to CIRA for funding this research, and to the atmos department for providing the ASCENT funding that supported my visit to ECMWF.

I extend a special thanks to Alan Geer. I have learned a lot from him about the knowledge of the global observing system and the data assimilation system in ECMWF. Thank Alan for his invaluable input to my paper and dissertation. Also, I greatly appreciate Alan for organizing and hosting my visit to ECWMF, which has been a truly enriching experience.

I would like to thank the van Leeuwen's research group members: Dandan, Michael, Kyle, Michelle, Ching-Shu and Charlotte. It has been a great pleasure meeting with them and learning from a wide array of diverse fields in research. Thanks to Christine, who has shown great care and support to me and to all the other Taiwanese students. Also thanks to the atmos department staff. All of them are so kind, patient, and helpful, ensuring my PhD journey progresses smoothly. I would also like to acknowledge all of my friends from the atmos department, and from Taiwan. Their company has made my life in Fort Collins truly memorable and have enriched my time here.

Finally, I would like to express my deepest appreciation to my family in Taiwan. Their boundless love and support have granted me the freedom to pursue my studies in the US without any concerns. I am immensely thankful for their unwavering encouragement, which has been the strongest driving force behind my academic journey.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

Weather forecasts have improved dramatically over the past decades, which can be attributed to two key factors: the advancement of the numerical modeling, and a better representation of the initial condition within the model. The importance of the latter is significant when the underlying dynamical system is chaotic. In such systems, a small perturbation in the initial condition can grow rapidly in a relatively short period of time. This phenomenon, also known as the "butterfly effect" (e.g., Lorenz and Haman, 1996), underscores the significance of a high-quality initial condition in generating reliable weather forecasts. In other words, the quality of the initial condition plays a crucial role in the weather forecast.

Unfortunately, generating a high-quality initial condition for the weather prediction model is a difficult task. Intuitively, one can gain information about the atmospheric state from observations. For example, the global observing system for the Earth includes the observations from surface stations, buoys, radiosondes, aircraft, ships, radars and satellites. Although the global observing system is huge, and becoming more comprehensive, the information derived solely from these observations is still far from sufficient for providing an initial condition for the model. One can imagine that it is almost impossible to have observations everywhere in the atmosphere and for all the atmospheric variables at a certain moment. In fact, a huge portion of the atmosphere is either unobserved or only indirectly observed. In other words, the global observing system only provides *partial* information of the atmosphere.

Another useful piece of the information, perhaps less intuitive, is the forecast from the weather model. One advantage of the forecast is that it contains *complete* information: it provides all the variables everywhere in the atmosphere at a certain time. Some may argue that the forecast produced by the model has errors, e.g., due to unrealistic parameterizations of physical processes that cannot be resolved in the weather model. Therefore, using the information from the model along with the observations can "contaminate" the observations. However, the above statement

is unfair, because all the observations have errors too. Actually, the model forecast can possess unique information about the dynamical system that can further enhance the information in the observations. Note that the argument is not about whether the model forecast or the observation is better. Instead, the argument is that they are both imperfect and have their own errors.

Now we have two major pieces of information for the atmosphere: one from the model forecast, and the other from the observations. They are both uncertain. The task to combine these two pieces of information to construct an initial condition, that respects both the model forecast and the observations, is called "Data Assimilation (DA)". Mathematically, this task can be described elegantly by Bayes theorem, which is the essence of DA. In the following, we will formally introduce DA, and explain definitions and notations in Section 1.1. Then, we will focus on the DA for atmospheric problems, identifying fundamental issues in the atmospheric DA in Section 1.2. Finally, we will present the research objectives and propose the corresponding research questions, which outlines the framework of this dissertation in Section 1.3.

1.1 Introduction to Data Assimilation (DA)

We first define the notations. Let $\mathbf{x} \in \mathbb{R}^{N_x}$ be the model state, which is the initial condition for the weather model that is the outcome from DA. The space where the model state lives is called the state space. Weather models discretize the atmosphere with "grid cells" (see e.g., Figure 1.1), and use numerous variables, e.g., temperature, wind, and humidity, within each grid cell to represent the continuous atmosphere. In other words, $\mathbf{x}$ is a very long vector containing all the variables within all the grid cells covering the whole atmosphere. For modern weather models, the dimension of the state space N_x can be as high as 10^9 to 10^{10}. Let $\mathbf{y} \in \mathbb{R}^{N_y}$ be the observation vector, living in the observation space. $\mathbf{y}$ is a concatenated vector containing all kinds of simulated observations from the global observing system. Typically N_y can be 10^7. The model state is related

to different types of the observations, using their own observation operators H_i. For example:

$$\mathbf{y} = H(\mathbf{x}) = \begin{bmatrix} H_1(\mathbf{x}) \\ H_2(\mathbf{x}) \\ \vdots \\ H_{N_y}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} H_{\text{temperature obs from weather station in Taipei}}(\mathbf{x}) \\ H_{\text{temperature obs from weather station in Fort Collins}}(\mathbf{x}) \\ \vdots \\ H_{\text{satellite radiance from 6.2 } \mu\text{m at (121°E, 25°N)}}(\mathbf{x}) \end{bmatrix} \tag{1.1}$$

where H_i's are the observation operators that simulate the observation from the model state. Different H_i have different functional form. It is worth noting that most of the observation operators H_i are not invertible. This is mainly because most of the observations are not located exactly at the model grid points. Therefore, H_i requires some interpolation of variables located at different model grid points. The interpolation operator is not one-to-one, and therefore any H_i that involves interpolation is not invertible. In other words, simply solving the state by "inverting the observation vector" is not a viable path. Furthermore, since the observations contain errors, this is also not desirable.

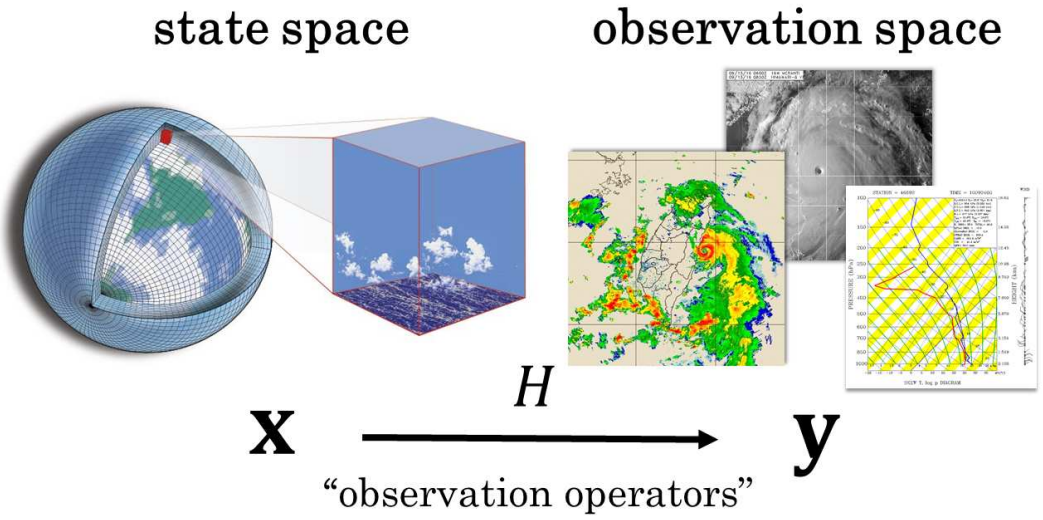

Figure 1.1: The demonstration of the state space and the observation space. $\mathbf{x}$ is the model state and $\mathbf{y}$ is simulated observation vector. H is a function, which contains all the observation operators (or forward operators), that maps an element in the state space to the observation space. Figure adapted from https://www.caltech.edu/about/news/new-climate-model-be-built-ground-84636

Recall that both the model state and the observations are uncertain. Therefore, we treat $\mathbf{x}$ and $\mathbf{y}$ as random vectors. Let $p_{\mathbf{x}}$ and $p_{\mathbf{y}}$ be the probability density functions (pdf) associated with $\mathbf{x}$ and $\mathbf{y}$, respectively. Instead of inverting the observation, the idea of DA is to use the information within the observation to constrain the uncertainty in the model forecast. Mathematically, we update the pdf from the model forecast $p_{\mathbf{x}}$ to the pdf after we see a realization of the observation $\mathbf{y} = \mathbf{y}_o$. Based on Bayes theorem, this updated pdf, i.e., the conditional pdf $p_{\mathbf{x}|\mathbf{y}}$ can be written as

$$p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}_o) = \frac{p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}_o|\mathbf{x})}{p_{\mathbf{y}}(\mathbf{y}_o)} p_{\mathbf{x}}(\mathbf{x}) \tag{1.2}$$

where $p_{\mathbf{x}}$ is called the prior (background) pdf, $p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}_o|\mathbf{x})$ is called the likelihood function (not a pdf, but a function of $\mathbf{x}$), and $p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}_o)$ is called the posterior (analysis) pdf. Since each pdf is clearly defined based on its input argument, for convenience, in the following we drop the subscript of each pdf and simplify Equation (1.2) as

$$p(\mathbf{x}|\mathbf{y}_o) = \frac{p(\mathbf{y}_o|\mathbf{x})}{p(\mathbf{y}_o)} p(\mathbf{x}) \tag{1.3}$$

Equation (1.3) is the foundation of DA. The combined information from the model forecast and the observations is recorded in the posterior pdf, which is the solution to the DA problem. Note that every DA methods can be derived from Equation (1.3), and different DA methods differ in their way to find and to represent the posterior pdf.

1.2 Fundamental challenges in atmospheric DA

Although Equation (1.3) is simple and elegant, many practical issues arise when one tries to apply it to atmospheric DA problems. The first question is: how to specify the prior pdf $p(\mathbf{x})$? Note that if we run the weather model to obtain a forecast at a certain time, we only get one state, instead of a distribution. The uncertainty comes from the fact that we acknowledge the forecast is imperfect, and the uncertainty in the forecast determines the distribution of $p(\mathbf{x})$. The uncertainty of the model output mainly comes from two sources: one is the model error, and the other is the

uncertainty of the model input, which is the model state in the previous time step. One can use, e.g., the Fokker-Planck equation to propagate the uncertainty. However, such evaluation of the pdf in a high dimensional (N_x is 10^9) and nonlinear system is unachievable due to the high computational cost.

The second question is: what is the likelihood function $p(\mathbf{y}_o|\mathbf{x})$? Note that the strategy here is that we first construct the function $p(\mathbf{y}|\mathbf{x})$ *before* DA. When an observation is seen $\mathbf{y} = \mathbf{y}_o$ in the *real-time* DA, we then evaluate the likelihood function $p(\mathbf{y} = \mathbf{y}_o|\mathbf{x})$ in Equation (1.3). So, how do we construct $p(\mathbf{y}|\mathbf{x})$? Observe that we can model the observation via the following relation:

$$\mathbf{y} = H(\mathbf{x}) + \varepsilon^o \tag{1.4}$$

where ε^o is a random vector that accounts for the uncertainties in the observation, which is called the observation error. Based on Equation (1.4), we can observe that $p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}|H(\mathbf{x}))$ and $p(\mathbf{y}|H(\mathbf{x}))$ is directly associated with the distribution of ε^o. Therefore, constructing $p(\mathbf{y}|\mathbf{x})$ is reduced to modeling the distribution of ε^o. However, modeling the distribution of ε^o is still not trivial.

Finally, assume we know the prior pdf $p(\mathbf{x})$ and the likelihood function $p(\mathbf{y}_o|\mathbf{x})$, we should be able to evaluate the posterior pdf $p(\mathbf{x}|\mathbf{y}_o)$ for each $\mathbf{x}$. However, such evaluation in high dimensional system is also unachievable due to the extremely demanding computational cost. Moreover, even if we can record the full posterior pdf $p(\mathbf{x}|\mathbf{y}_o)$, how are we going to use $p(\mathbf{x}|\mathbf{y}_o)$ to initialize the model?

Despite all these challenges, DA is still possible, and in fact it has been quite successful in modern weather forecasting. The modern DA methods can be broadly classified into two types: the variational methods and the ensemble-based methods. The key for the success is that they make assumptions to approximate Equation (1.3) so that efficient algorithms can be developed even in a high-dimensional system. In the remainder of this Section, we will briefly review the variational method in Section 1.2.1, the ensemble-based method in Section 1.2.2. The remaining issues related to the common assumptions will be discussed in Section 1.2.3.

1.2.1 Variational methods

In the variational method (e.g., 3D-Var or 4D-Var), it is assumed that the prior pdf $p(\mathbf{x})$ is Gaussian. If the observation operator H is linear and the observation error pdf ε^o is Gaussian, then it can be shown that the likelihood function $p(\mathbf{y}_o|\mathbf{x})$ is also Gaussian. This leads to a Gaussian posterior pdf $p(\mathbf{x}|\mathbf{y}_o)$ from Equation (1.3). If we define a cost-function $J(\mathbf{x})$ by taking the negative of log posterior pdf, i.e.,

$$J(\mathbf{x}) = -\log p(\mathbf{x}|\mathbf{y}_o) \tag{1.5}$$

then $J(\mathbf{x})$ is quadratic in $\mathbf{x}$. Since there exists efficient algorithms to minimize a quadratic cost-function, we can find the mode (also the mean for Gaussian) of the posterior pdf by minimizing $J(\mathbf{x})$ in Equation (1.5):

$$\mathbf{x}_a = \underset{\mathbf{x}}{\operatorname{argmin}}\ J(\mathbf{x}) \tag{1.6}$$

where $\mathbf{x}_a$ is taken as the analysis, or the outcome from the variational method. In other words, we seek the most likely state given the model forecast and the observation in the variational method. When H is non-linear, one can find an approximated mode by iteratively finding the minimum state for a sequence of quadratic cost-functions, which is obtained by locally linearizing H. This is known as the "incremental form". See Section 5.2.1 for details.

In the variational method, we choose the model forecast as the mean of the Gaussian prior. The prior covariance $\mathbf{B}$ is typically constructed based on the climatology of the dynamical system, and therefore $\mathbf{B}$ is *static*, or *flow-independent*. The observation error ε^o is assumed to be Gaussian and its pdf can be either pre-specified or estimated from the statistics of model simulations and the observations (see Section 4.1, 4.2.2 for further details). In practice, we solve Equation (1.6) for "one analysis state", which is used as the initial condition to obtain a forecast for the next DA cycle.

The caveats of the variational method is that it does not provide explicit error estimates of the posterior $p(\mathbf{x}|\mathbf{y}_o)$. In principle, the posterior covariance can be inferred from the inverse of the Hessian at the mode, assuming H is linear. However, when H is non-linear, the posterior is non-

Gaussian and the variational method can not provide an accurate uncertainty estimate for $p(\mathbf{x}|\mathbf{y}_o)$. Moreover, assuming a *static* $\mathbf{B}$ can be problematic especially for extreme weather events, whose error might not be well-captured based on the climatological statistics. The inaccurate $p(\mathbf{x})$ due to the static prior covariance can also lead to a poor estimate for $p(\mathbf{x}|\mathbf{y}_o)$.

1.2.2 Ensemble-based methods

The ensemble-based methods are based on the idea of Monte Carlo methods. A Monte Carlo method is a way to approximate integrals over a pdf for any the nonlinear system, such as the mean and the covariance. Specifically, one first randomly samples the initial pdf, and let the nonlinear system forecast the state of each sample. Then, use the distribution of the forecast samples as an approximation of the prior pdf $p(\mathbf{x})$. In other words, one avoids explicitly solving the Fokker-Planck equation by approximating the solution using finite samples (also called *ensemble members* or *particles*). Still, it can be difficult to infer the mathematical form of the pdf given a finite number of samples. Therefore, an easy and useful way is to assume the pdf is Gaussian, and infer its 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). In this way, the prior covariance changes with the flow of the dynamical system, i.e., it is *flow-dependent*. This is the advantage of the ensemble-based methods over the variational methods.

Similar to the variational method, if assuming linear H and Gaussian-distributed observation errors ε^o, one can derive efficient ways to represent the Gaussian posterior pdf $p(\mathbf{x}|\mathbf{y}_o)$. Unlike the variational method which only seeks the mode of the posterior, the ensemble-based methods find ways to sample $p(\mathbf{x}|\mathbf{y}_o)$. It can be shown that based on the linear and Gaussian assumptions, there exists efficient linear transformations (e.g., Anderson, 2001; Bishop et al., 2001; Whitaker and Hamill, 2002; Tippett et al., 2003, etc) that can transform the sample from the prior to a sample from the posterior. In addition to a deterministic transform, there are also stochastic methods like the stochastic EnKF (e.g., Burgers et al., 1998; Houtekamer and Mitchell, 1998; van Leeuwen,

2020) that can efficiently sample $p(\mathbf{x}|\mathbf{y}_o)$ in a high dimensional space under these assumptions. In practice, the ensemble DA methods find an updated "ensemble of states", which is used as the initial conditions for the ensemble forecast. Then, the distribution of the ensemble forecast determines the prior distribution in the next DA cycle.

Although there are methods to slightly relax the linear assumption for H, for moderately non-linear H these assumptions can still be inappropriate. Similarly, a non-linear H can lead to non-Gaussian likelihood $p(\mathbf{y}_o|\mathbf{x})$, which results in a non-Gaussian posterior $p(\mathbf{x}|\mathbf{y}_o)$. The ensemble methods based on linear and Gaussian assumptions are unable to provide an accurate representation of a non-Gaussian $p(\mathbf{x}|\mathbf{y}_o)$.

1.2.3 Remaining issues

As discussed in the previous two sections, the Gaussian and linear assumptions are the two major assumptions for many DA methods that are applied to atmospheric problems. These methods rely on the assumptions that all the pdfs are (close to) Gaussian so efficient methods can work in a high-dimensional space. We note that the linear assumption is entangled with the Gaussian assumption: if H is non-linear, then the likelihood $p(\mathbf{y}_o|\mathbf{x})$ is non-Gaussian, resulting in a non-Gaussian posterior $p(\mathbf{x}|\mathbf{y}_o)$. Although some of these methods can be extended to slightly non-linear H, they still cannot well represent $p(\mathbf{x}|\mathbf{y}_o)$ for a fully non-linear and non-Gaussian problem.

Nevertheless, are the non-linearities and the non-Gaussianities really important for atmospheric DA problem? One might argue that, with a more comprehensive global observing system and a more advanced model that can resolve subgrid processes, the model trajectory can be better constrained so that the non-linearities can be reduced, which in turns reduces the non-Gaussianities. However, this argument is only partially correct. We can expect the synoptic-scale features become better constrained and more Gaussian, while on the other hand, the smaller and convective-scale features can introduce a new source of non-Gaussianities (e.g., Bocquet et al., 2010). This is because at the convective scale, variables such as vertical velocity and hydrometeors can present non-Gaussian errors to become more important. In addition, a huge portion of the global observ-

ing system nowadays is from the satellite observations, which are often non-linearly related to the model state. However, many of these observations, especially associated with cloud and precipitations, are still discarded due to the deficiencies in DA by many operational centers (Geer et al., 2018). In fact, these remotely-sensed observations can also present non-Gaussian observation errors ε^o, which adds further complexity to the problem. Therefore, fundamental issues arising from the non-linearities and the non-Gaussianities in atmospheric DA still exist, and will persist, which can limit our utilization of the valuable information contained within the observations.

1.3 Research Objectives

The goal of this dissertation is to investigate the issues arising from the non-Gaussianities and the non-linearities in DA, and propose possible solutions to these problems. The dissertation can be broadly divided into two parts. In the first part of the dissertation, we aim to explore the following research question:

Research Question 1

Is there a method that can efficiently draw samples from a non-Gaussian posterior pdf in a high-dimensional system?

Recall that efficient methods to draw samples from the posterior pdf in high-dimensional systems are optimal for a Gaussian posterior. However, what if the posterior pdf is non-Gaussian, e.g., due to the non-linear observation operator or the non-Gaussian observation error? In Chapter 2, we will explore one of the new developments in DA methodology, the particle flow filter (PFF), which holds promises of solving the fully non-linear and non-Gaussian DA problem, while it has only been tested in low-order models. Therefore, we aim to explore the behavior of the PFF in a high-dimensional system.

Note that in the weather forecast problem, the dimension of the model is not only high, but extremely high. Therefore, the scalability of a DA algorithm is very important for a real atmospheric DA implementation. In Chapter 3, we aim to address the following question:

Research Question 2

Following research question 1, can we use this method to sample a non-Gaussian posterior pdf in an atmospheric problem?

In the second part, we turn our attention to the observation errors ε^o. The pdf of ε^o plays a crucial role in DA as it determines the shape of the likelihood function $p(\mathbf{y}_o|\mathbf{x})$. The pdf of ε^o has been assumed to be Gaussian in DA, mainly due to two reasons. First, we do not have method to estimate the full pdf of ε^o, except for its second moment. In other words, we assume a Gaussian ε^o because we do not know other higher-moment information. Therefore, in Chapter 4, we aim to answer the following scientific question:

Research Question 3

Is it possible to estimate the full distribution of the observation error ε^o even when it is non-Gaussian? What are the distributions of the observation errors for all-sky satellite microwave radiances (which form an important part of the global observing system)? When do these observations exhibit non-Gaussian errors?

The second reason is that most of the DA methods used in the operational weather forecasting centers are developed based on the assumption of Gaussian ε^o. Although the PFF or other advanced filters can naturally adopt non-Gaussian ε^o, replacing the core DA algorithm in an operational center is certainly an arduous mission. Instead, the operational center prefers an *incremental* change to improve their system. Therefore, in Chapter 5, we aim to seek a practical method that can efficiently incorporate non-Gaussian ε^o into the variational method, which is a commonly used DA method in many operational centers. Specifically, the research question is:

Research Question 4

How do we incorporate non-Gaussian observation errors (e.g., from the outcome of research question 3) into a variational method in general? How do the non-Gaussianities affect the accuracy of forecasts?

In the remainder of this dissertation, we will investigate the research questions 1-4 in Chapter 2-5, respectively. We have a more comprehensive literature review regarding each research question in the first section in each chapter. In Chapter 6, we will provide our answers to these research questions, and then conclude the dissertation with outlook and future implications.

Chapter 2

A particle flow filter for high dimensional system applications[1]

2.1 Background and motivation

As mentioned in Chapter 1, current operational DA methods for weather forecasting are based on linear and Gaussian assumptions. One method that is not limited by any constraints in the prior distribution and linearity assumptions is particle filter (PF). PF is a method that has potential to sample any non-parametric posterior pdfs. The PF is also an ensemble-based method, which represents the pdf using particles (ensemble members) with weights. For example, the prior pdf $p(\mathbf{x})$ can be represented by the particles $\mathbf{x}_i$ with weight w_i:

$$p(\mathbf{x}) = \sum_{i=1}^{N_p} w_i \delta(\mathbf{x} - \mathbf{x}_i) \tag{2.1}$$

where $\delta(\mathbf{x} - \mathbf{x}_i)$ is the Delta function. In the standard PF, the posterior pdf is represented by updating the weights of the particles from the prior pdf, which can be seen based on Bayes theorem in Equation (1.3):

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}_o) &= \frac{p(\mathbf{y}_o|\mathbf{x})}{p(\mathbf{y}_o)} \sum_{i=1}^{N_p} w_i \delta(\mathbf{x} - \mathbf{x}_i) \\ &= \sum_{i=1}^{N_p} \frac{p(\mathbf{y}_o|\mathbf{x}_i)}{p(\mathbf{y}_o)} w_i \delta(\mathbf{x} - \mathbf{x}_i) \\ &= \sum_{i=1}^{N_p} \tilde{w}_i \delta(\mathbf{x} - \mathbf{x}_i) \end{aligned} \tag{2.2}$$

[1]This chapter has been published in *Quaterly Journal of the Royal Meteorological Society*: Hu, C.-C., and P. J. van Leeuwen, 2021: A particle flow filter for high-dimensional system applications. *Quarterly Journal of the Royal Meteorological Society*, **147 (737)**, 2352–2374, but with minor adaptations in Section 2.1 to avoid overlap with Chapter 1.

where $\tilde{w}_i = \frac{p(\mathbf{y}_o|\mathbf{x}_i)}{p(\mathbf{y}_o)} w_i$ is the updated weight given by the standard PF. Despite its non-parametric advantage, the standard PF suffers from the issue of "weight degeneracy" in a high-dimensional system (e.g., see Snyder et al., 2008; van Leeuwen, 2009, for details). Specifically, one particle gets a weight close to one, while the other particles have negligible weights. Therefore, we lose the variations from the particles when weights degenerate. We require an unfeasible number of particles to avoid weight degeneracy in the standard PF. This problem has made the application of the standard PF into geoscience models difficult.

Nevertheless, much progress has been made to deal with the weight degeneracy problems, including the usage of proposal densities (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 (e.g., Chorin and Tu, 2009; 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 of equal weight for all the time. The particles are iteratively transformed from the prior to the posterior in state space, without the need to resample or to re-weight 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) have successfully assimilated nonlinear observations, or observations with bimodal likelihood in the 40 variables 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 PPF to high dimensional nonlinear problems, and to compare the performance of the PPF to the Local Ensemble Kalman Filter (LETKF), which has been widely used in the 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 chapter is organized as follows. Section 2.2 introduces the theoretical background for the PPF, and the effect of the different kernels will be discussed. Section 2.3 describes the data assimilation experiments used to evaluate the performance of the LETKF and the PFF. Section 2.4 compares the experiment results between a well-tuned LETKF and the PPF with linear and nonlinear observations. Section 2.5 conducts sensitivity experiments to see the effect of different settings in the PFF. Section 2.6 summarizes the results and discusses future applications to geosciences models.

2.2 Methodology

2.2.1 Introduction to the Particle Flow Filter (PFF)

The particle flow filter (PFF) (e.g., 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:

$$\frac{d}{ds}\mathbf{x}_s = \mathbf{f}_s(\mathbf{x}_s),\ s \in [0, \infty] \tag{2.3}$$

with

$$q_0(\mathbf{x}) = p(\mathbf{x})$$

$$q_\infty(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}_o)$$

where $\mathbf{x}_s$ is a state at pseudo time s, $\mathbf{f}_s$ is the particle flow (the transformation) which 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 $q_\infty(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}_o)$ 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:

$$KL(q_s) = \int q_s(\mathbf{x}) \log \left(\frac{q_s(\mathbf{x})}{q_\infty(\mathbf{x})} \right) d\mathbf{x} \tag{2.4}$$

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 $\mathbb{R}^{N_x} \times \mathbb{R}^{N_x} \rightarrow \mathbf{M}_{N_x \times N_x}(\mathbb{R})$, where N_x is the dimension of the system and $\mathbf{M}_{N_x \times N_x}(\mathbb{R})$ is a matrix of size $N_x \times N_x$. 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 in Section A.1 for the derivations), as:

$$\mathbf{f}_s(\cdot) = \mathbf{D} \int q_s(\mathbf{x}) \left\{ \mathbf{K}(\mathbf{x}, \cdot) \nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y}_o) + \nabla_{\mathbf{x}} \cdot \mathbf{K}(\mathbf{x}, \cdot) \right\} d\mathbf{x} \tag{2.5}$$

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 $q_s(\mathbf{x})$:

$$q_s(\mathbf{x}) = \frac{1}{N_p} \sum_{i=1}^{N_p} \delta(\mathbf{x} - \mathbf{x}_s^i) \tag{2.6}$$

and therefore the flow field at pseudo time s in Equation (2.5) can be written as

$$\mathbf{f}_s(\mathbf{x}) = \mathbf{D}\sum_{i=1}^{N_p}\left\{\mathbf{K}(\mathbf{x}_s^i,\mathbf{x})\nabla_{\mathbf{x}_s^i}\log p(\mathbf{x}_s^i|\mathbf{y}_o) + \nabla_{\mathbf{x}_s^i}\cdot\mathbf{K}(\mathbf{x}_s^i,\mathbf{x})\right\} \tag{2.7}$$

Note that we have replaced the dot in Equation (2.5) with the state $\mathbf{x}$ whose particle flow is being evaluated in Equation (2.7). 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 we discretize Equation (2.3) in pseudo time, and use Equation (2.7), to evaluate the movement of any state at pseudo time s to pseudo time $s+\Delta s$:

$$\mathbf{x}_{s+\Delta s} = \mathbf{x}_s + \frac{\Delta s}{N_p}\mathbf{D}\sum_{i=1}^{N_p}\left\{\mathbf{K}(\mathbf{x}_s^i,\mathbf{x}_s)\nabla_{\mathbf{x}_s^i}\log p(\mathbf{x}_s^i|\mathbf{y}_o) + \nabla_{\mathbf{x}_s^i}\cdot\mathbf{K}(\mathbf{x}_s^i,\mathbf{x}_s)\right\} \tag{2.8}$$

with small Δs.

The first term on the RHS of Equation (2.7) 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 $K(\mathbf{x}_s^i,\mathbf{x})$ 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 $\nabla_{\mathbf{x}_s^i}\log p(\mathbf{x}_s^i|\mathbf{y}_o)$. 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 2.1a shows an example of how the kernel works.

The second term on the RHS of Equation (2.7) acts as a repelling term. To demonstrate its repelling nature, assume we have a scalar kernel:

$$\mathbf{K}(\mathbf{x}_s^i,\mathbf{x}) = K(\mathbf{x}_s^i,\mathbf{x})\mathbf{I}_{N_x} \tag{2.9}$$

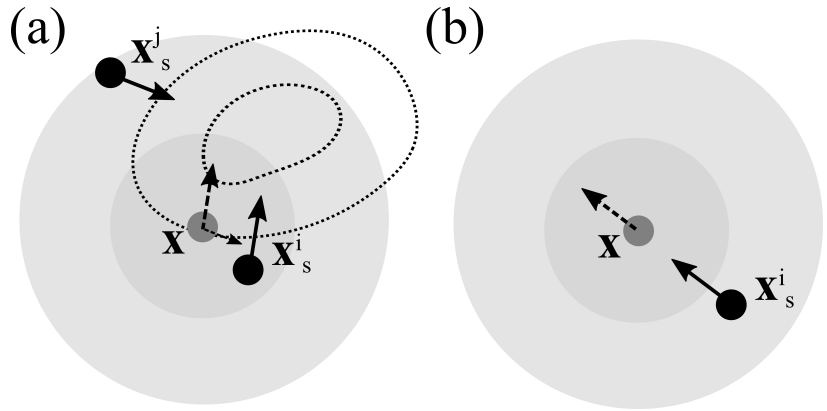

Figure 2.1: The demonstration of how the particle flow is represented by particles. The black dots are the states of particles and the grey dot is the state x at which the particle flow is being evaluated. The shaded grey circles are the kernel, with darker shading representing greater value. (a) The weighted average of the gradient of the logarithm of posterior (the first term in Equation (2.7)). 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 (2.7)). 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.

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

$$K(\mathbf{x}_s^i, \mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x}_s^i - \mathbf{x})^T \mathbf{A}(\mathbf{x}_s^i - \mathbf{x})\right) \tag{2.10}$$

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

$$\nabla_{\mathbf{x}_s^i} \cdot \mathbf{K}(\mathbf{x}_s^i, \mathbf{x}) = \begin{bmatrix} \frac{\partial}{\partial x_1^i} K(\mathbf{x}_s^i, \mathbf{x}) \\ \frac{\partial}{\partial x_2^i} K(\mathbf{x}_s^i, \mathbf{x}) \\ \vdots \\ \frac{\partial}{\partial x_{N_x}^i} K(\mathbf{x}_s^i, \mathbf{x}) \end{bmatrix} = \nabla_{\mathbf{x}_s^i} K(\mathbf{x}_s^i, \mathbf{x}) \tag{2.11}$$

Figure 2.1b shows an example. The gradient of 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 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 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 (2.7), the gradient of the logarithm of posterior (see Section 2.2.2) and of the kernel (see Section 2.2.3) need to be determined, which will be discussed in the following sections.

2.2.2 The logarithm of the posterior in 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 in Equation (1.3), we have

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y}_o) = \nabla_{\mathbf{x}} \log p(\mathbf{y}_o|\mathbf{x}) + \nabla_{\mathbf{x}} \log p(\mathbf{x}) \tag{2.12}$$

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

$$\log p(\mathbf{y}_o|\mathbf{x}) \propto -\frac{1}{2}\|\mathbf{y}_o - H(\mathbf{x})\|_{\mathbf{R}} \tag{2.13}$$

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

$$\nabla_{\mathbf{x}} \log p(\mathbf{y}_o|\mathbf{x}) = \mathbf{H}^T \mathbf{R}^{-1} (\mathbf{y}_o - H(\mathbf{x})) \tag{2.14}$$

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

$$\mathbf{H} := \frac{dH}{d\mathbf{x}}(\mathbf{x}) \tag{2.15}$$

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

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}) = \mathbf{B}^{-1} (\mathbf{x} - \mathbf{x}_b) \tag{2.16}$$

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.

2.2.3 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

$$\mathbf{K}(\mathbf{x}, \mathbf{z}) = K(\mathbf{x}, \mathbf{z})\mathbf{I}_{N_x} \tag{2.17}$$

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

$$K(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{z})^T \mathbf{A}(\mathbf{x} - \mathbf{z})\right) \tag{2.18}$$

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

$$\mathbf{A} = (\alpha \mathbf{B})^{-1} \tag{2.19}$$

To contrast with a kernel that will be introduced later, we will refer the kernel in Equation (2.17) as the scalar kernel. The divergence of the scalar kernel is

$$\nabla_{\mathbf{x}} \cdot \mathbf{K}(\mathbf{x}, \mathbf{z}) = \begin{bmatrix} \frac{\partial}{\partial x_1} K(\mathbf{x}, \mathbf{z}) \\ \frac{\partial}{\partial x_2} K(\mathbf{x}, \mathbf{z}) \\ \vdots \\ \frac{\partial}{\partial x_{N_x}} K(\mathbf{x}, \mathbf{z}) \end{bmatrix} = -\mathbf{A}^T(\mathbf{x} - \mathbf{z})K(\mathbf{x}, \mathbf{z}) \tag{2.20}$$

Pulido and van Leeuwen (2019) have shown that the scalar kernel works well for the 40-variables Lorenz 96 system with 20 observations (50% of the system is observed). Lu et al. (2019) has shown that 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 1000 variables with 250 observations (25% of the system is observed), problems arise 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:

$$\mathbf{K}(\mathbf{x},\mathbf{z}) = \operatorname{diag}\left(\left[K_{(1)}(\mathbf{x},\mathbf{z}), K_{(2)}(\mathbf{x},\mathbf{z}), \cdots, K_{(N_x)}(\mathbf{x},\mathbf{z})\right]\right) \tag{2.21}$$

where

$$K_{(a)}(\mathbf{x},\mathbf{z}) = K_{(a)}\left(x_{(a)}, z_{(a)}\right) = \exp\left(-\frac{1}{2}\frac{\left(x_{(a)} - z_{(a)}\right)^2}{\alpha {\sigma_{(a)}}^2}\right) \tag{2.22}$$

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

$$\mathbf{x} = \begin{bmatrix} x_{(1)} \\ x_{(2)} \\ \vdots \\ x_{(N_x)} \end{bmatrix} \tag{2.23}$$

and $\sigma_{(a)}$ is the standard deviation 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 2.5.1. The divergence of the matrix-valued kernel is

$$\begin{aligned} \nabla_{\mathbf{x}_s^i} \cdot \mathbf{K}\left(\mathbf{x}_s^i, \mathbf{x}\right) &= \begin{bmatrix} \frac{\partial}{\partial x_1^i} K_{(1)}\left(x_{s,(1)}^i, x_{(1)}\right) \\ \frac{\partial}{\partial x_2^i} K_{(2)}\left(x_{s,(2)}^i, x_{(2)}\right) \\ \vdots \\ \frac{\partial}{\partial x_{N_x}^i} K_{(N_x)}\left(x_{s,(N_x)}^i, x_{(N_x)}\right) \end{bmatrix} \\ &= \begin{bmatrix} -\left(\frac{x_{s,(1)}^i - x_{(1)}}{\alpha {\sigma_{(1)}}^2}\right) K_{(1)}\left(x_{s,(1)}^i, x_{(1)}\right) \\ -\left(\frac{x_{s,(2)}^i - x_{(2)}}{\alpha {\sigma_{(2)}}^2}\right) K_{(2)}\left(x_{s,(2)}^i, x_{(2)}\right) \\ \vdots \\ -\left(\frac{x_{s,(N_x)}^i - x_{(N_x)}}{\alpha {\sigma_{(N_x)}}^2}\right) K_{(N_x)}\left(x_{s,(N_x)}^i, x_{(N_x)}\right) \end{bmatrix} \end{aligned} \tag{2.24}$$

We note that the most important difference between the scalar (Equation 2.17) and the matrix-valued (Equation 2.21) 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 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 due to the unobserved components, and hence value of the kernel can still be small 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 2.20) is proportional to the value of 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 what we have 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 2.24) 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 2.2 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 (Figs. 2.2a,b). This can occur when the observation has a strong impact (or highly correlated) on the update of the unobserved variable. When the unobserved component converges much slower than observed component in Figs. 2.2c,d, the repelling force of the scalar kernel remains small even though the observed components are very close (Fig. 2.2d), 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 (Fig. 2.2c), 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 1000-dimensional Lorenz 96 system detailed in the Section 2.3. Figure 2.3 shows the posterior marginal distribution of the variable $x_{(19)}$ (unobserved component) and $x_{(20)}$ (observed component) after the first data assimilation update. This is similar to the case in Figs. 2.2c,d: the matrix-valued kernel is able to keep particles away from each other, preventing the collapse in the observed component (Fig. 2.3a), while the particles collapse for $x_{(20)}$ using the scalar kernel (Fig. 2.3b).

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 neighbours. We did not need that in the experiments below, but it is an interesting path of research for the future.

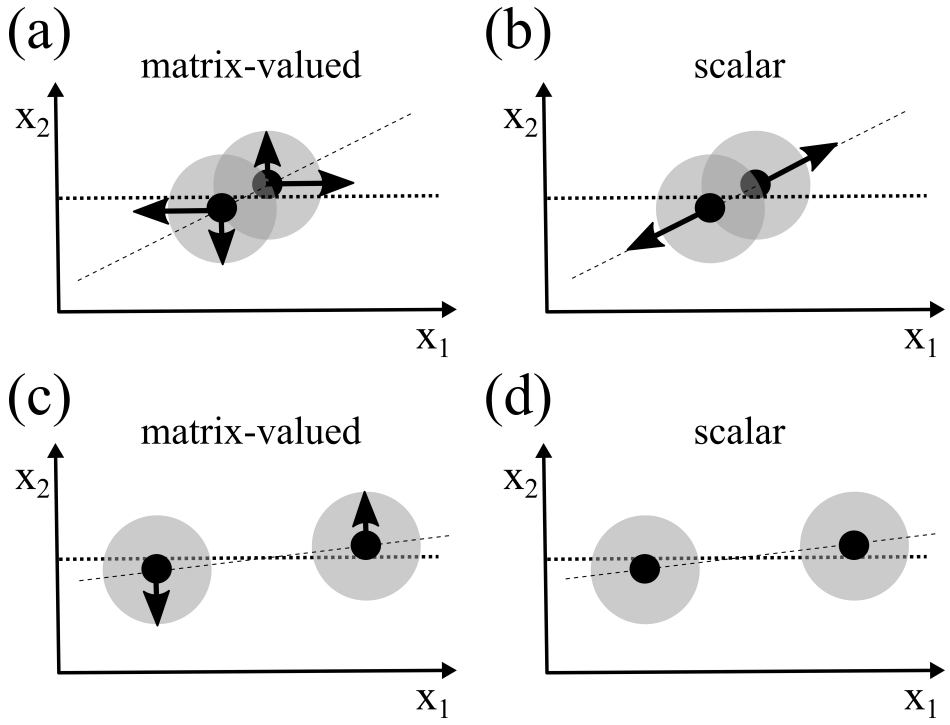

Figure 2.2: The comparison of the divergence of kernel between (a)(c) the matrix-valued kernel and (b)(d) the scalar kernel. The shading is the scalar kernel (which is assumed to be an independent Gaussian with equal variance), the black dots are the particles, and the black arrows are the repelling forces resulting from the divergence of kernel. x_1 denotes an unobserved component, and x_2 is an observed component. (a)(b) show the divergence for both kernels when the convergence rate of observed and unobserved components is similar, leading to similar repelling forces. (c)(d) show the 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.

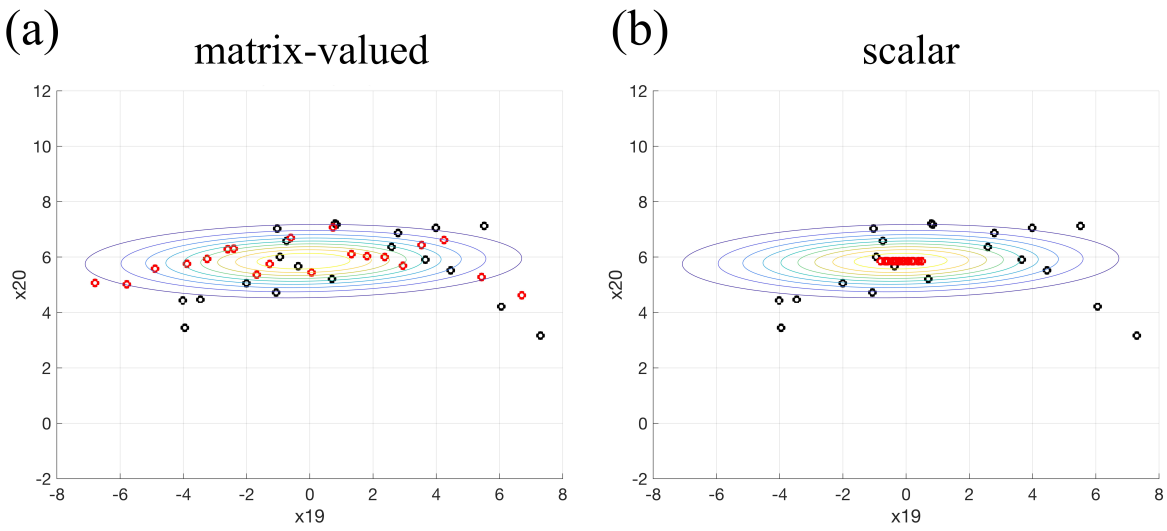

Figure 2.3: The prior and posterior marginal distribution of the variable $x(19)$ (unobserved component) and $x_{(20)}$ (observed component) before and after the first data assimilation update (t=20) with linear observation operator defined in Equation (2.31), using the (a) matrix-valued kernel (b) scalar 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.

2.2.4 Implementation of the PFF

Algorithm 1 summarizes the steps for implementing the PFF (Note that the algorithm 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). 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

Algorithm 1 A simple algorithm for the PFF implementation

Main code:

Assume a Gaussian prior for the following:

$\overline{\mathbf{x}}_0 \leftarrow \frac{1}{N_p}\sum_{i=1}^{N_p}\mathbf{x}_0^i$ ▷ ensemble mean $(N_x \times 1)$

$\mathbf{X} \leftarrow \left[\mathbf{x}_0^1 - \overline{\mathbf{x}}_0, \mathbf{x}_0^2 - \overline{\mathbf{x}}_0, \ldots, \mathbf{x}_0^{N_p} - \overline{\mathbf{x}}_0\right]$ ▷ 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 (2.29), (2.30) $(N_x \times N_x)$

$s = 0$ ▷ the pseudo time

repeat

  for $i = 1, \cdots, N_p$ **do**

    $\mathbf{y}^i \leftarrow H(\mathbf{x}_s^i)$ ▷ modeled observation $(N_y \times 1)$

    $\mathbf{H}^i \leftarrow \frac{dH}{d\mathbf{x}}(\mathbf{x}_s^i)$ ▷ linearized observation operator $(N_y \times N_x)$

    $\nabla \log p(\mathbf{x}_s^i|\mathbf{y}_o) \leftarrow \mathbf{H}^{i^T}\mathbf{R}^{-1}(\mathbf{y}_o - \mathbf{y}^i) - \mathbf{B}^{-1}(\mathbf{x}_s^i - \overline{\mathbf{x}}_0)$ ▷ grad of log posterior $(N_x \times 1)$

  end for

  for $d = 1, \cdots, N_x$ **do**

    for $i = 1, \cdots, N_p$ **do**

      $f_{s\,(d)}^i \leftarrow 0$ ▷ 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)

      $\frac{\partial p}{\partial x}_{(d)}^{i} \leftarrow \mathbf{e}_{(d)}^T\nabla \log p(\mathbf{x}_s^i|\mathbf{y}_o)$ ▷ the d^{th} component of $\nabla \log p(\mathbf{x}_s^i|\mathbf{y}_o)$ (1×1)

      for $j = 1, \cdots, N_p$ **do**

        $x_{(d)}^j \leftarrow \mathbf{e}_{(d)}^T\mathbf{x}_s^j$ ▷ the d^{th} component of $\mathbf{x}_s^j$ (1×1)

        $K_{(d)}^{i,j} \leftarrow \exp\left(-\frac{1}{2}\frac{\left(x_{(d)}^i - x_{(d)}^j\right)^2}{\alpha\mathbf{B}_{d,d}}\right)$ ▷ the kernel in Equation (2.22) (1×1)

        $\frac{\partial K}{\partial x}_{(d)}^{i,j} \leftarrow -\left(\frac{x_{(d)}^i - x_{(d)}^j}{\alpha\mathbf{B}_{d,d}}\right)K_{(d)}^{i,j}$ ▷ the grad of kernel in Equation (2.24) (1×1)

        $I_{f\,(d)}^i \leftarrow I_{f\,(d)}^i + \frac{1}{N_p}\left(K_{(d)}^{i,j}\frac{\partial p}{\partial x}_{(d)}^{i} + \frac{\partial K}{\partial x}_{(d)}^{i,j}\right)$ ▷ the integral in Equation (2.7) (1×1)

      end for

    end for

  end for

  for $i = 1, \cdots, N_p$ **do**

    $\mathbf{f}_s^i \leftarrow \mathbf{B}\mathbf{I}_f^i$ ▷ multiply the preconditioner $\mathbf{D} = \mathbf{B}$ in Equation (2.7) $(N_x \times 1)$

    $\mathbf{x}_s^i \leftarrow \mathbf{x}_s^i + \Delta s\mathbf{f}_s^i$ ▷ update the ensemble of states in Equation (2.8) $(N_x \times 1)$

  end for

  $s \leftarrow s + 1$

until stopping criterion met

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, e.g. by producing unrealistic gravity waves, which will affect the quality of the forecast.

The pseudo time step (the time step for the iterations in Equation 2.8) Δs should be small enough to prevent instability of the iterative procedure. We use an adaptive scheme to determine the pseudo time step Δs in order 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 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.

2.3 Experimental Design

We compare the effect of the PFF and the local Ensemble Transform Kalman Filter (LETKF) (Hunt et al., 2007) by applying both methods to the Lorenz 96 model (Lorenz and Emanuel, 1998). The equation for the N_x-dimensional Lorenz 96 model is

$$\frac{dx_{(a)}}{dt} = \left(x_{(a+1)} - x_{(a-2)}\right) x_{(a-1)} - x_{(a)} + F \tag{2.25}$$

where $a = 1, \cdots, N_x$ and $x_{(a)}$ is the a-th component of the state $\mathbf{x}$ defined in Equation (2.23). We set N_x to 1000 and F to 8. The 4^{th} order Runge-Kutta scheme is used, and the time resolution is

$\Delta t = 0.01$. The initial condition is set as

$$x_{(a)}(t=0) = \begin{cases} F, & \text{if} \quad \mathrm{mod}(a,5) \neq 0 \\ F+1, & \text{if} \quad \mathrm{mod}(a,5) = 0 \end{cases} \tag{2.26}$$

and the model is integrated for 1000 time steps to generate chaotic behavior. At $t = 1000$, $N_p = 20$ 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 $t = 1000$ 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 hours in atmospheric models). The magnitude of the observational error ε depends on the observation operators. The observation is taken at every 4th 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 1500 timesteps.

In order to obtain a more statistically reliable comparison between the PFF and the LETKF, the experiment 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 1500 timesteps for these 9 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 localization on the inverse of the observation error covariance $\mathbf{R}_i$ when updating the i^{th} grid point:

$$\mathbf{R}_i^{-1} = \mathbf{R}^{-1} \circ \mathbf{C}_i \tag{2.27}$$

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

$$\mathbf{C}_i = \mathrm{diag}\left(\left[\exp\left\{-\left(\frac{d_{i,1}}{r_{in}}\right)^2\right\}, \exp\left\{-\left(\frac{d_{i,2}}{r_{in}}\right)^2\right\}, \ldots, \exp\left\{-\left(\frac{d_{i,N_y}}{r_{in}}\right)^2\right\}\right]\right) \tag{2.28}$$

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:

$$\mathbf{B} \leftarrow \mathbf{B} \circ \mathbf{C} \tag{2.29}$$

where

$$\mathbf{C} = [c_{i,j}]_{N_x \times N_x}, \quad c_{i,j} = \exp\left\{-\left(\frac{i-j}{r_{in}}\right)^2\right\} \tag{2.30}$$

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:

$$H_{\text{linear}}(\mathbf{x}) = \begin{bmatrix} x_{(4)} \\ x_{(8)} \\ \vdots \\ x_{(N_x)} \end{bmatrix}_{N_y \times 1} \tag{2.31}$$

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:

$$H_{abs}(\mathbf{x}) = \begin{bmatrix} \left|x_{(4)}\right| \\ \left|x_{(8)}\right| \\ \vdots \\ \left|x_{(N_x)}\right| \end{bmatrix}_{N_y \times 1} \tag{2.32}$$

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 $\varepsilon = 0.5$. We will also test the behavior of the methods for an exponential observation operator, given by:

$$H_{\text{exp}}(\mathbf{x}) = \begin{bmatrix} e^{\frac{x_{(4)}}{6}} \\ e^{\frac{x_{(8)}}{6}} \\ \vdots \\ e^{\frac{x_{(N_x)}}{6}} \end{bmatrix}_{N_y \times 1} \tag{2.33}$$

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

operator, given by

$$H_{\text{square}}(\mathbf{x}) = \begin{bmatrix} x_{(4)}^2 \\ x_{(8)}^2 \\ \vdots \\ x_{(N_x)}^2 \end{bmatrix}_{N_y \times 1} \tag{2.34}$$

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 $\varepsilon = 1$.

2.4 Results

The performance of the PFF is tested in a sequential data assimilation experiment as described in Section 2.3. In order 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.

2.4.1 Linear observation operator

In order 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:

$$RMSE\_X(t) = \sqrt{\frac{1}{N_x}\sum_{a=1}^{N_x}\left(\bar{x}_{(a)}(t) - x_{t_{(a)}}(t)\right)^2} \tag{2.35}$$

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

experiments, which is

$$RMSE\_X\_OBS(t) = \sqrt{\frac{1}{|OBS|} \sum_{(a)\in OBS} \left(\bar{x}_{(a)}(t) - x_{t_{(a)}}(t)\right)^2} \tag{2.36}$$

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 non-observed variables is:

$$RMSE\_X\_NOOBS(t) = \sqrt{\frac{1}{|NOOBS|} \sum_{(a)\in NOOBS} \left(\bar{x}_{(a)}(t) - x_{t_{(a)}}(t)\right)^2} \tag{2.37}$$

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

Figure 2.4 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 2.4a 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 2.4b shows that the RMSE of the observed variables reduces 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 Fig.2.4a.

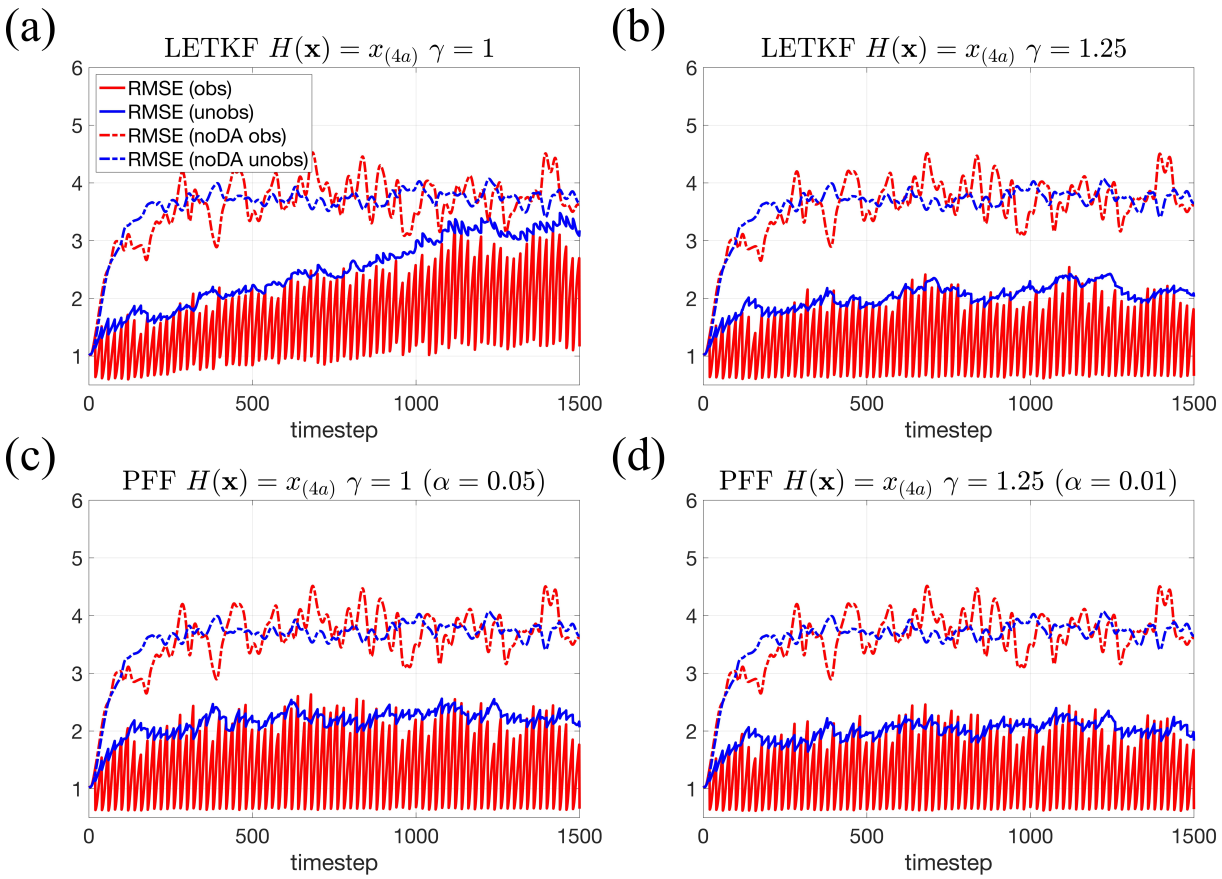

Figure 2.4: The root-mean-square error (RMSE) of the noDA ensemble (i.e., the ensemble with same initial conditions but without data assimilation, dashed lines) and with data assimilation (solid lines) using (a) the LETKF without inflation $\gamma = 1$ (b) the LETKF with inflation factor $\gamma = 1.25$ (c) the PPF without inflation $\gamma = 1$ and with kernel width $\alpha = 0.05$ (d) the PPF with inflation factor $\gamma = 1.25$ and with kernel width $\alpha = 0.01$. The RMSE are calculated for the observed variables (red lines), and the unobserved variables (blue lines).

With a proper choice of the kernel width α in Equation (2.22), we are able to have a stable RMSE of the observed variables for the PPF either with or without inflation of the prior (Figs.2.4c,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 decrease. Compare the performance of the LETKF with inflated prior (Fig.2.4b) and the PFF with inflated prior (Fig.2.4d), we find their RMSE of the observed variables are 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 2.4 suggests that the performance of the PPF is overall comparable to a well-tuned the 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 to the truth at the observation times are shown in Fig.2.5. It is shown that the rank histogram for the LETKF without inflation (Fig.2.5a) is close to U-shape, suggesting that the ensemble may be either biased against the truth or under-dispersive. Based on Fig.2.4a, 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 (Fig.2.5b). The rank histogram from the PFF are quite close to the uniform distribution, despite that the ensemble is slightly over-dispersive (Figs.2.5c-d). The rank histogram for the unobserved variables are 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 set up used here, both the PPF and the LETKF assume

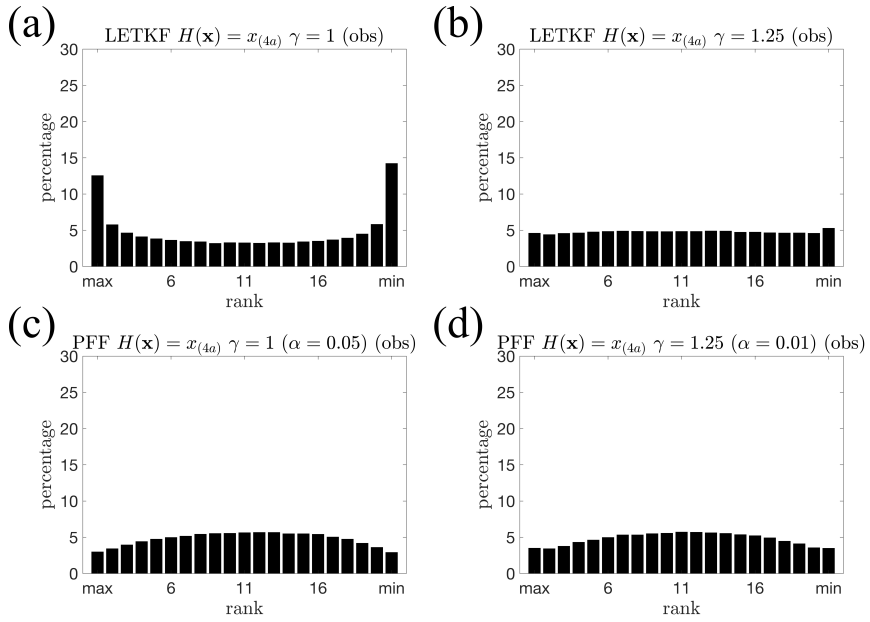

Figure 2.5: The rank histogram of the observed variables for the prior compared to the truth at the observation times for the linear observation. The ensemble is from (a) the LETKF without inflation $\gamma = 1$ (b) the LETKF with inflation factor $\gamma = 1.25$ (c) the PPF without inflation $\gamma = 1$ and with kernel width $\alpha = 0.05$ (d) the PPF with inflation factor $\gamma = 1.25$ and with kernel width $\alpha = 0.01$.

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 2.5.1) for the PFF can make the posterior wider, in order 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 prior and the observations. Note that when there is evidence that the prior is under-dispersive, we can still inflate the prior for the PFF. This suggests that the PFF is more flexible than the LETKF.

2.4.2 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 for the behavior of the ensemble, and so a RMSE in the model state may not be a useful measure of performance. An example

is that when the posterior is a multi-modal distribution. This happens for the square observation $H(\mathbf{x}) = {x_{(4a)}}^2$, where the observation operator is not one-to-one even if we confine the domain to the subset of the observed variable. In order 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 multi-modal. The RMSE of the observed variables defined in the observational space is:

$$RMSE\_Y(t) = \sqrt{\frac{1}{|OBS|} \sum_{(a)\in OBS} \left(\bar{y}_{(a)}(t) - y_{t_{(a)}}(t)\right)^2} \tag{2.38}$$

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

$$\bar{y}_{(a)} = \frac{1}{N_p} \sum_{i=1}^{N_p} H\left(x^i_{(a)}\right) \tag{2.39}$$

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

$$y_{t_{(a)}} = H\left(x_{t,(a)}\right) \tag{2.40}$$

Figure 2.6 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 to the noDA ensemble (Fig.2.6a). For the exponential operator, the results from LETKF are not very stable: it is 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 (Fig.2.6c). Similar results can be found for the square operator for the LETKF (Fig.2.6e). In contrast, the performance of the PFF for these three nonlinear observations are all very good and stable (Figs.2.6b,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 $t = 1500$.

For the behavior of the unobserved variables, the LETKF only shows an improvement in the RMSE for the absolute value observation before t=300 (Fig.2.7a), for the exponential observation before t=1000 (Fig.2.7c). After that, the RMSE from the LETKF is almost indistinguishable from the noDA ensemble for the absolute value operator (Fig.2.7a), and becomes even worse for the other two observational operators (Figs.2.7c,e). In contrast, the RMSE from the PFF shows an improvement for all the observational operators over the noDA ensemble (Figs.2.7b,d,f). It is worth to note that for the PFF, the RMSE of the observed variables in model space for the absolute value (Fig.2.7b) and square observations (Fig.2.7f) at observation times are quite variable, while their values are very stable when evaluated in the observation space (Figs.2.6b,f). This demonstrates the problem of using 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. Figures 2.8a,b show that for the absolute value operator, both of 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 away 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 (Figs.2.8c,e). This is a result of biased mean as is evident in Figs.2.7c,e. This suggests that the dependence of the linearized observational operator on the state is a major

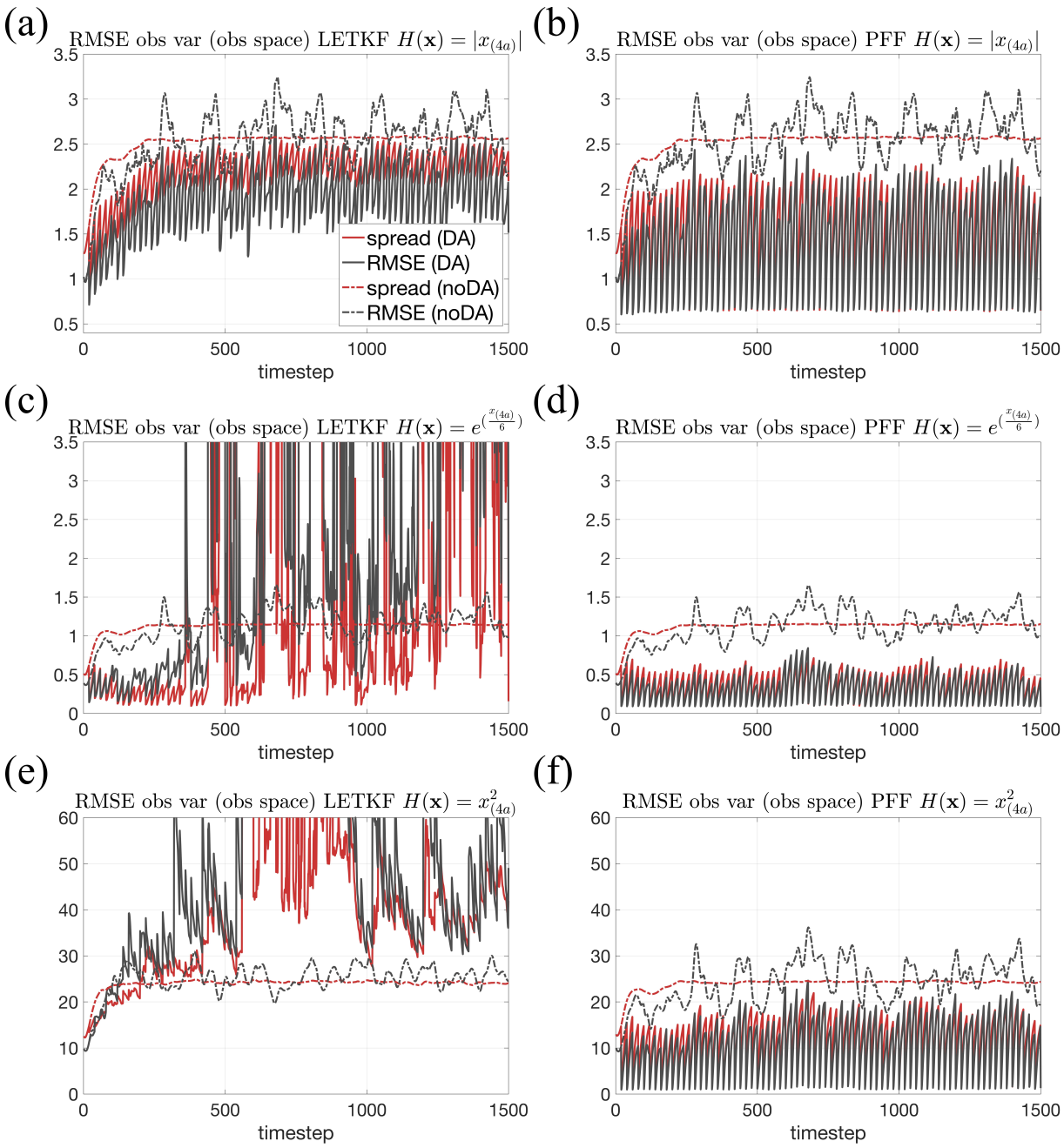

Figure 2.6: The RMSE (red) and the ensemble spread (gray) of the observed variables defined in the observational space for the nonlinear observations. The dashed lines are from the noDA ensemble, and the solid lines are from the (a)(c)(e) LETKF (b)(d)(f) PFF. The observation operator is (a)(b) the absolute value operator (c)(d) the exponential operator, and (e)(f) the square operator.

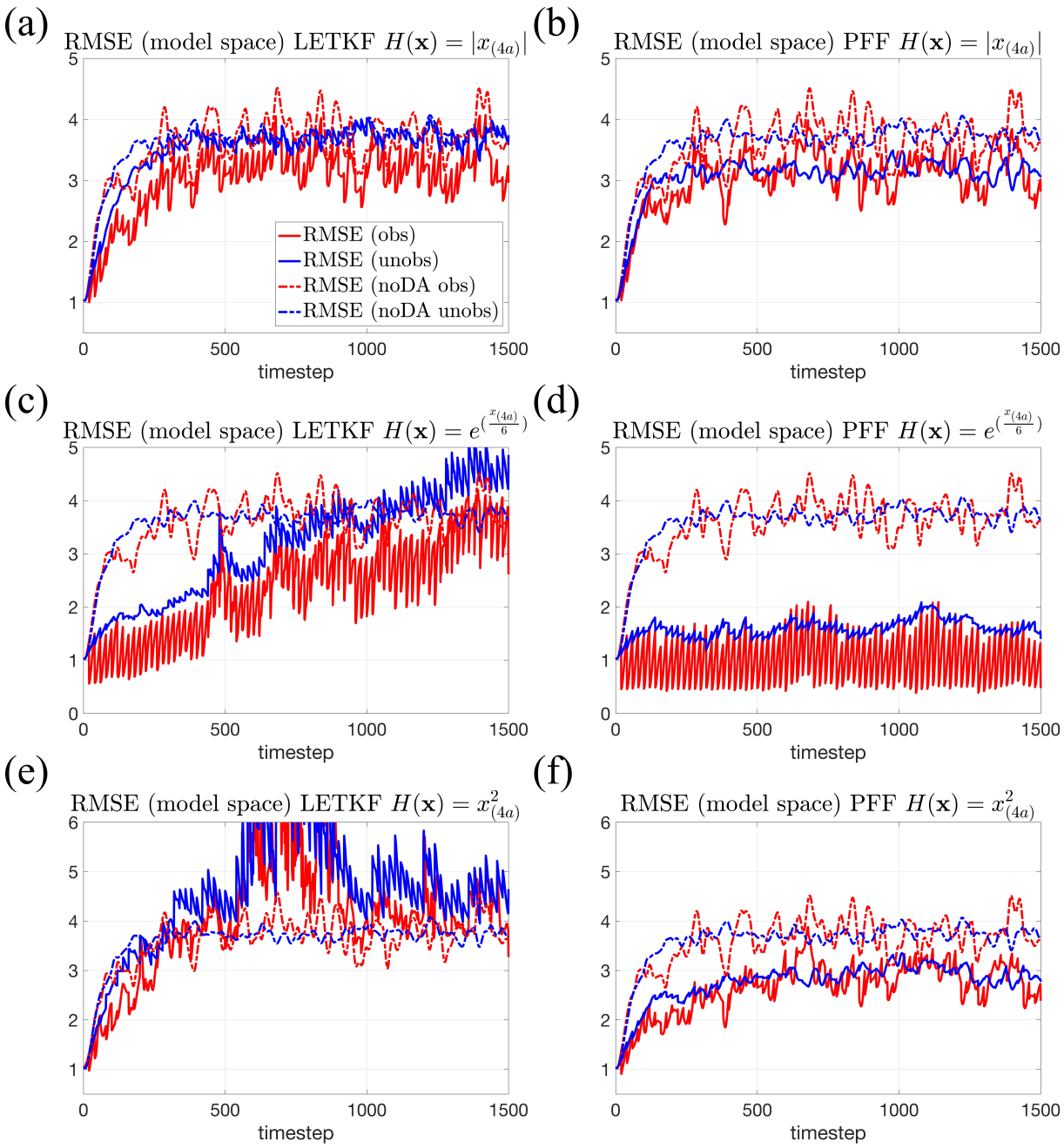

Figure 2.7: The same as Fig.2.3, but for the nonlinear observations. The solid lines are from the (a)(c)(e) LETKF (b)(d)(f) PFF. The observation operator is (a)(b) the absolute value operator (c)(d) the exponential operator, and (e)(f) the square operator.

reason for the biased mean in the observational space for the LETKF. For the PFF, Figs.2.8b,d,f show that the rank histograms for all the nonlinear operators are close to the uniform distribution, although slightly over-dispersive for the exponential operator.

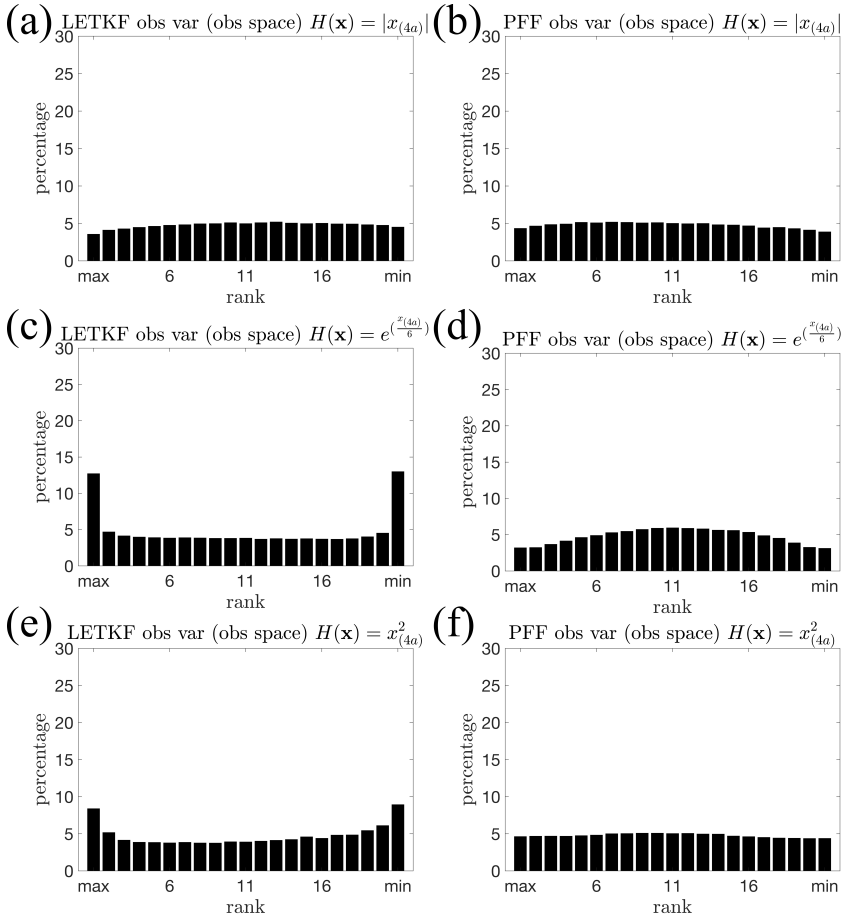

Figure 2.8: The same as Fig.2.5, but for the observed variables in the observational space for the nonlinear observations. The ensemble is from the (a)(c)(e) LETKF (b)(d)(f) PFF. The observation operator is (a)(b) the absolute value operator (c)(d) the exponential operator, and (e)(f) the square operator.

To demonstrate the ability of PFF to generate multimodal posterior, we compare the time series of one of the observed variable $x_{(108)}$ with square observations during $t = 250 - 500$ in Fig.2.9. Note that there can be two 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 as the second solution (blue dots in Fig.2.9). 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 of the methods demonstrate an improvement of the variable evolution compared to the noDA ensemble (Fig.2.9). 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 (Fig.2.9b). In contrast, for the PFF, the ensemble is able to follow both the true and the second solution at the same time (Fig.2.9c).

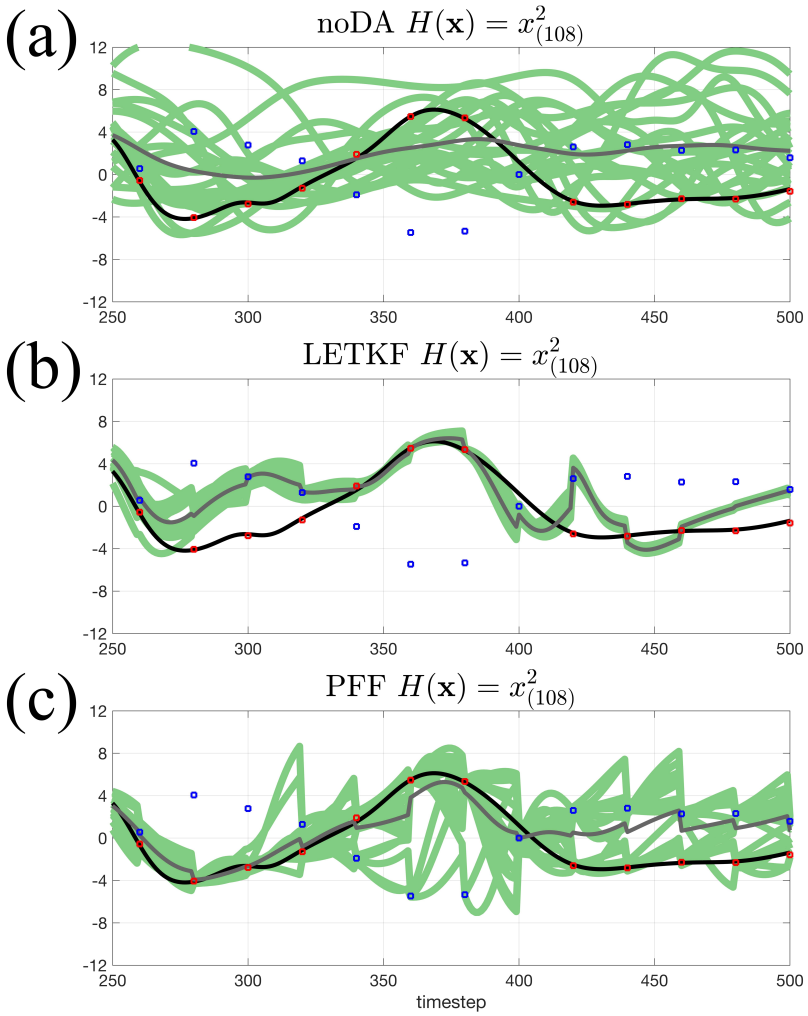

Figure 2.9: The time series of the observed variable $x_{(108)}$ during $t = 250 - 500$. The black line is the truth, the green lines are the ensemble members, and the gray line is the ensemble mean. The square observation is assimilated into the system. The red dots are the solution with the same sign of the truth corresponding to the observations, and the blue dots are the computational solution with the opposite sign of the truth corresponding to the observations. The ensemble is from (a) noDA (b) LETKF (c) PFF.

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 2.10 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 (Fig.2.10b). Compared to the LETKF, the ensemble from the PFF has a larger spread, but follows the truth much better (Fig.2.10c). 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 Fig.2.7f). The result suggests 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 (Fig.2.9c), 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 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 (Fig.2.9b). 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 much more modes when considering all components together. In this sense, the behavior of the PFF is quite remarkable.

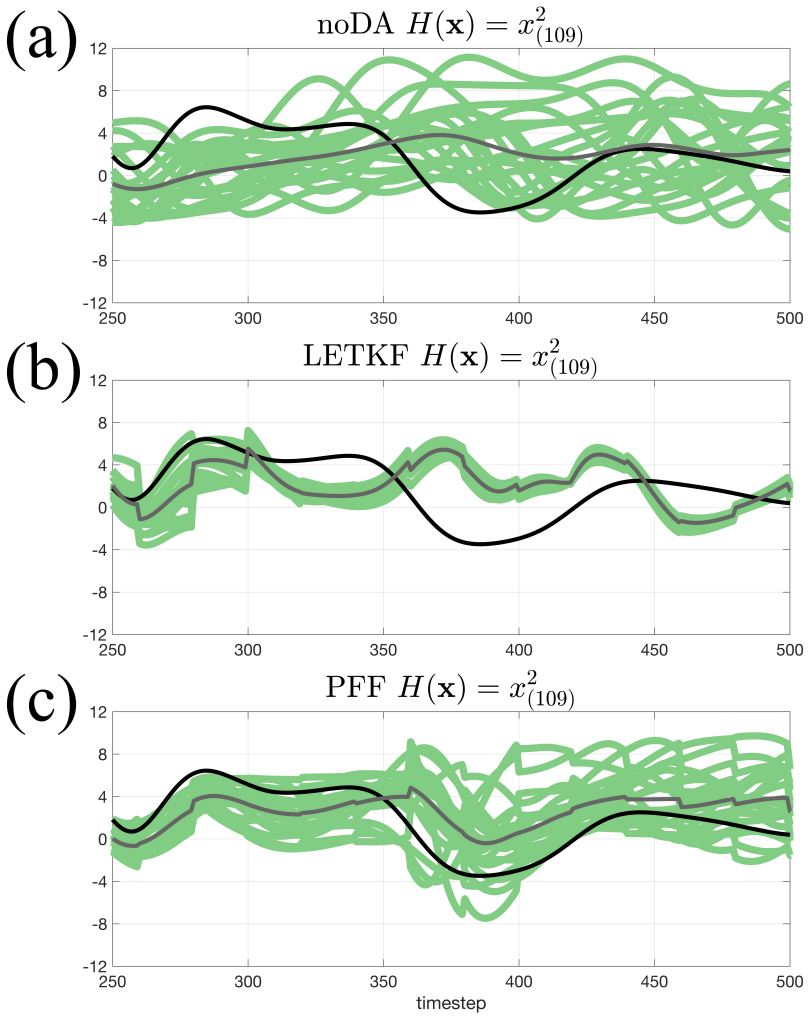

Figure 2.10: The same as Fig.2.9 but for the unobserved variable $x_{(109)}$. The ensemble is from (a) noDA (b) LETKF (c) PFF.

2.5 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 2.3 are used.

2.5.1 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 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 limited number of particles, the particles are sparse in the space. Therefore, using a small α with limited number of particles may cause the particles to almost collapse to the mode (i.e., become too close to the mode). Due to the above reasons, when we have more particles, we would like to make α smaller, but not too small, in order 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 2.11 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 Fig.2.11) 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.

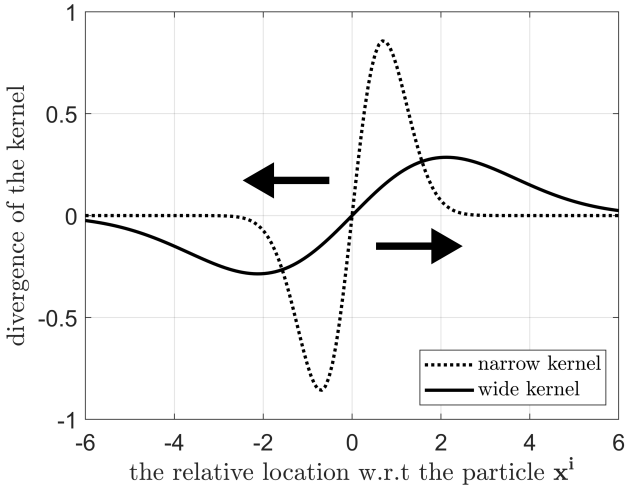

Figure 2.11: Demonstration of the effect of kernel width on the divergence of the kernel (repelling force). The system is assumed to be one-dimensional. The x axis is the relative location with respect to one of the particle, and the y axis is the divergence of the kernel that is applied on the state whose particle flow is evaluated. The dashed (solid) line represents a kernel width small (large) kernel width. The arrow represents the direction of the repelling force.

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 2.4 except for the kernel width. Figure 2.12 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 (Fig.2.12b). Figure 2.13 shows the total RMSE of the ensemble for 500 timesteps. It is found that when 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 2.14 shows the rank histogram of the prior for the observed variables at the observation times. For the smaller kernel width (Figs.2.14a,b), the rank histogram suggests the ensemble may be either biased against the truth or under-dispersive. For larger kernel width (Figs.2.14c,d), the rank histogram suggests that the ensemble is slightly over-dispersive. 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.

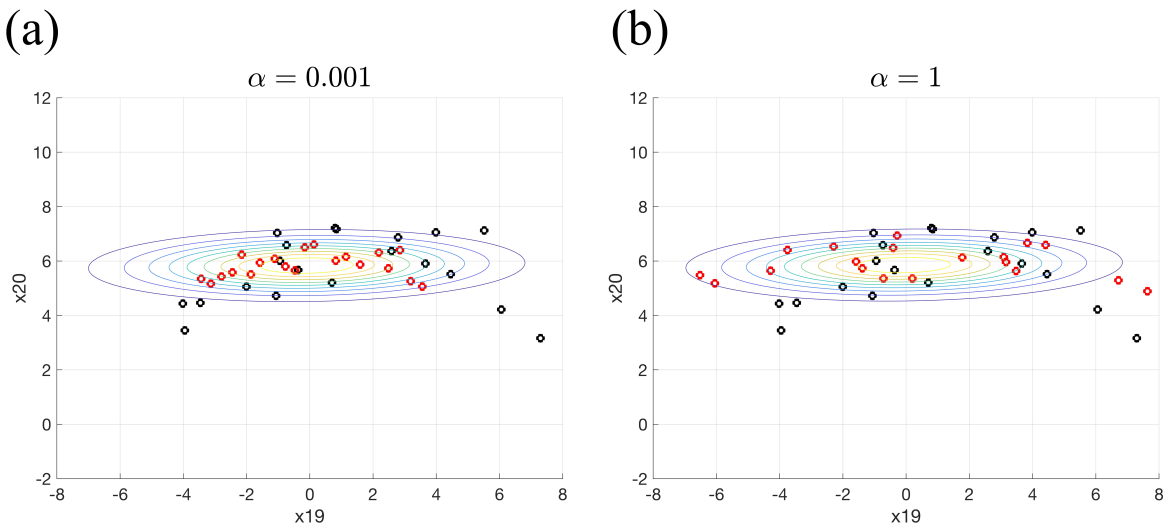

Figure 2.12: The same as Fig.2.3, but the kernel used here are both matrix-valued kernels yet with different kernel widths. (a) A narrow kernel width $\alpha = 0.001$ (b) A wide kernel width $\alpha = 1$.

2.5.2 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 2.15 illustrates this sensitivity by showing the total RMSE of the

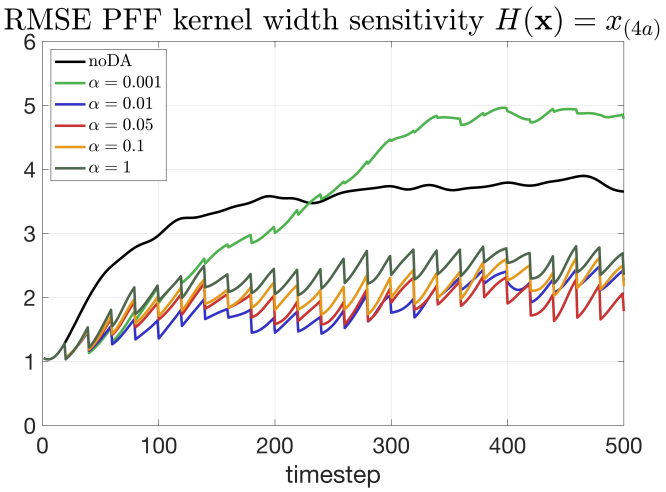

Figure 2.13: The total RMSE of the ensemble from the PFF, assimilating linear observations, with different kernel widths.

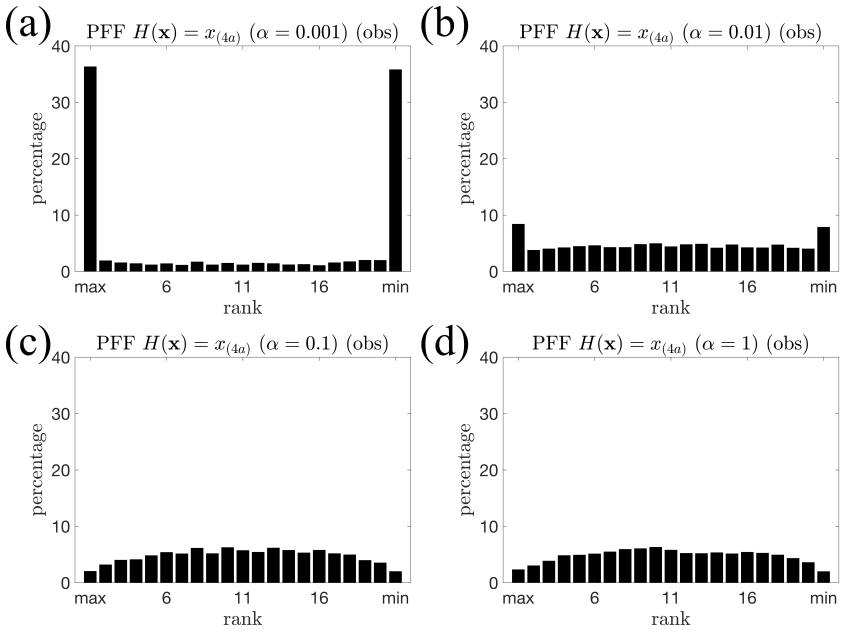

Figure 2.14: The same as Fig.2.5, but for the PFF with different kernel width. (a) $\alpha = 0.001$ (b) $\alpha = 0.01$ (c) $\alpha = 0.1$ (d) $\alpha = 1$. For the experiment with $\alpha = 0.05$, see Fig.2.5c.

ensemble over 500 timesteps. It is found that once the number of iteration is over 50, the RMSE is quite similar for the linear observation case (Fig.2.15a). On the other hand, the RMSE becomes quasi-steady only when the number of iteration is more than 200 for the exponential observation (Fig.2.15b). 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 is expected to be more and more pronounced if we further extend the integration time. On the other hand, we can expect less 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.

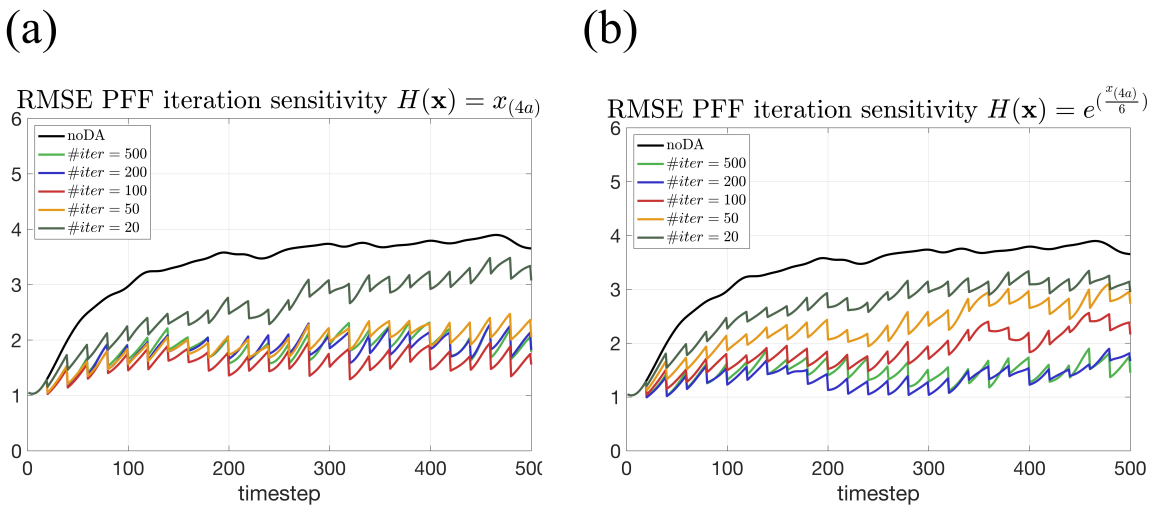

Figure 2.15: The same as Fig.2.13, but for a different number of iterations for the PFF using (a) the linear observation (b) the exponential observation.

2.5.3 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, in order 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 (e.g., 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 e.g. 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 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 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 that many of the marginal distributions for the observed variables in the prior are 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 4 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 $\mathbf{E}$ 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:

$$p\left(\mathbf{x}_0\right) = \frac{1}{N_p} \sum_{i=1}^{N_p} N\left(\mathbf{x}_0^i, \mathbf{E}\right) \tag{2.41}$$

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 $\mathbf{E}$ 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

$$\begin{aligned} \mathrm{Cov}\left(\mathbf{x}_0\right) &= \mathbf{E} + \frac{1}{N_p} \sum_{i=1}^{N_p} \left(\mathbf{x}_0^i - \overline{\mathbf{x}_0}\right) \left(\mathbf{x}_0^i - \overline{\mathbf{x}_0}\right)^T \\ \overline{\mathbf{x}_0} &= \frac{1}{N_p} \sum_{i=1}^{N_p} \mathbf{x}_0^i \end{aligned} \tag{2.42}$$

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

$$\mathbf{B} = \mathrm{Cov}\left(\left\{\mathbf{x}_0^i\right\}_{i=1}^{N_p}\right) = \frac{1}{N_p - 1} \sum_{i=1}^{N_p} \left(\mathbf{x}_0^i - \overline{\mathbf{x}_0}\right) \left(\mathbf{x}_0^i - \overline{\mathbf{x}_0}\right)^T \tag{2.43}$$

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

$$\mathbf{E} = \frac{\mathbf{B}}{N_p} \tag{2.44}$$

Experiments with the Gaussian mixture in Equation (2.41) show that the covariance for each component $\mathbf{E}$ is too narrow, making the update of each particle very limited (not shown). We can improve this update by making $\mathbf{E}$ wider through multiplication with a scalar, so that each component is able to interact with each other. However, we note that making $\mathbf{E}$ wider is to some extent equivalent to inflating the prior. We should not make $\mathbf{E}$ too wide so that the information of the prior will be lost. Nevertheless, we can still tune the width of $\mathbf{E}$ 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 in 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.

2.6 Concluding remarks

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-Leibner 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 only have 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 particles 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 only have 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. In order to tackle this problem, a new matrix-valued Gaussian kernel is proposed in this study. The proposed kernel is a diagonal matrix with 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 1000-dimensional Lorenz 96 system, with only 25% of the system is 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 less 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 pre-conditioner 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 Gaussian mixture prior does not show significant improvement over the PFF with a Gaussian prior, mainly related with 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 into 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 the need of 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 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 so-called "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.

Chapter 3

An implementation of the particle flow filter in an atmospheric model

3.1 Background and motivation

Although the theoretical background of the particle flow filter (PFF) is complete (see Chapter 2), developing a functional and efficient PFF algorithm within a high-dimensional DA system is not trivial. Recall that the dimension of the weather forecasting model is very high (e.g., 10^9). Therefore, we require a scalable algorithm for PFF, especially one that can run in parallel.

As will be detailed later, we develop an algorithm that is consistent with the idea of the Two-Step Ensemble Filtering (TSEF). TSEF was originally proposed by Anderson (2003) as an efficient and mathematically consistent algorithm to implement the ensemble Kalman filter. One of the key properties that makes TSEF a scalable algorithm is that each of the model variables can be updated independently from each observation. This allows TSEF to run in parallel. TSEF is the core DA algorithm in the Data Assimilation Research Testbed (DART), which is a community data assimilation software for geosciences applications. Furthermore, TSEF was recently generalized by Grooms (2022) based on Bayes theorem, which further opens up new research opportunities for non-Gaussian DA methods (e.g., Anderson, 2022, 2023).

Therefore, the goal of this chapter is to devise an scalable algorithm for PFF that can run in parallel. We choose to develop the PFF algorithm in DART, which is called PFF-DART, in order to facilitate the implementation of PFF into atmospheric models. The remainder of this chapter is organized as follows. In Section 3.2, we will describe the details of TSEF, its connection to Bayes theorem, and the PFF-DART algorithm. Then, we conduct an Observing System Simulation Experiment (OSSE) to compare the performance of the PFF-DART with the Ensemble Adjustment Kalman Filter (EAKF) (Anderson, 2001). The experiment setup of the OSSE will be detailed in

Section 3.3. When assimilating linear and Gaussian observations, we will demonstrate PFF-DART can produce stable and comparable results to EAKF in a year-long cycling DA setup in Section 3.4. When assimilating non-linear and/or non-Gaussian observations, we will show the capability of PFF-DART to yield better results than EAKF in Section 3.5. Finally, we provide a summary of this work and discuss the future prospects and potential advancements of the PFF-DART in Section 3.6.

3.2 Methodology

In this section, we will present a new approach to implementing and testing the particle flow filter (PFF) in a high-dimensional geophysical model. We utilize the Data Assimilation Research Testbed (DART), which will be described in Section 3.2.1, to facilitate the implementation of PFF in geophysical models. In order to develop an efficient and parallelizable PFF code in DART, we revise the PFF based on a *generalized* two-step ensemble filtering algorithm, called PFF-DART, as described in Sections 3.2.2-3.2.3.

3.2.1 Data Assimilation Research Testbed (DART)

In order to test PFF in a high-dimensional geophysical model, we choose to code PFF into the Data Assimilation Research Testbed (DART) (Anderson et al., 2009; DAReS, 2022). DART is an open-source community ensemble data assimilation (EDA) software package that is carefully engineered to efficiently implement EDA schemes in many geophysical models. DART also has many commonly used EDA techniques, such as localization and inflation that can be coupled with different EDA schemes. In addition, common geophysical observation operators are available. Therefore, we can utilize many existing features of DART and facilitate the implementation of PFF in a geophysical model.

3.2.2 The two-step ensemble filtering (TSEF) algorithm

The core EDA algorithm in DART is based on a two-step ensemble filtering (TSEF) algorithm (Anderson, 2003). One major advantage of the TSEF algorithm is that it is efficient and parallelizable. Specifically, each state variable can be updated independently from each observation, allowing for parallel update for each state variable. The TSEF algorithm is applied for the assimilation of each scalar observation, based on the assumption of independent observation errors. In the *original* TSEF, the first step updates the ensemble in the observation space based on a Bayesian method, and the second step updates the ensemble in the model space. Specifically, in the first step, an EDA method is applied to obtain posterior samples in the observation space $H(x)^i_{posterior}$ ($i = 1, ..., N_p$), and the ensemble of increments in the observation space is recorded as $\Delta H(x)^i = H(x)^i_{posterior} - H(x)^i_{prior}$. In the second step, $\Delta H(x)^i$ is used as the predictor to calculate the ensemble increments in the state space $\Delta x^i = x^i_{posterior} - x^i_{prior}$ based on the simple linear regression between $H(x)_{prior}$ and x_{prior}.

Assume the observation is linear, and univariate ensemble Kalman filters are used as the EDA method in the first-step, Anderson (2003) proved that the *original* TSEF algorithm is equivalent to the full multivariate ensemble Kalman filters, while the TSEF algorithm avoids the explicit computation of the covariance matrix of the full model state and its inverse. Grooms (2022) generalized the TSEF algorithm by making the connection between TSEF and Bayes theorem. The *generalized* TSEF provides the theoretical background for more advanced EDA methods or sampling techniques to be applied in the TSEF algorithm. The *generalized* TSEF works as follows. First, a variable $\mathbf{z}$ is introduced, which satisfies

$$p(y_o|\mathbf{x}, \mathbf{z}) = p(y_o|\mathbf{z}) \tag{3.1}$$

where y_o is the scalar observation and $\mathbf{x}$ is the model state. Then, based on Bayes theorem in Equation (1.3) we can re-write the posterior pdf as

$$p(\mathbf{x}|y_o) = \int p(\mathbf{x}, \mathbf{z}|y_o)d\mathbf{z} = \int \frac{p(y_o|\mathbf{x}, \mathbf{z})p(\mathbf{x}, \mathbf{z})}{p(y_o)}d\mathbf{z} \tag{3.2}$$

Based on Equations (1.3) and (3.1), we can further write the posterior pdf as

$$\begin{aligned} p(\mathbf{x}|y_o) &= \int \frac{p(y_o|\mathbf{z})p(\mathbf{x}, \mathbf{z})}{p(y_o)}d\mathbf{z} \\ &= \int \frac{p(y_o|\mathbf{z})p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(y_o)}d\mathbf{z} \\ &= \int \left[\frac{p(y_o|\mathbf{z})p(\mathbf{z})}{p(y_o)}\right] p(\mathbf{x}|\mathbf{z})d\mathbf{z} \\ &= \int p(\mathbf{z}|y_o)p(\mathbf{x}|\mathbf{z})d\mathbf{z} \end{aligned} \tag{3.3}$$

If we draw N_p samples $\mathbf{z}^i$ from $p(\mathbf{z}|y_o)$ in Equation (3.3), i.e.,

$$p(\mathbf{z}|y_o) = \frac{1}{N_p}\sum_{i=1}^{N_p} \delta(\mathbf{z} - \mathbf{z}^i) \tag{3.4}$$

Then, Equation (3.3) becomes

$$p(\mathbf{x}|y_o) = \frac{1}{N_p}\sum_{i=1}^{N_p} p(\mathbf{x}|\mathbf{z}^i) \tag{3.5}$$

Equation (3.5) suggests that, if we draw one sample from each of $p(\mathbf{x}|\mathbf{z}^i)$, then the collection of these N_p draws together will be a sample from $p(\mathbf{x}|y_o)$. Therefore, the *generalized* TSEF can be summarized as follows:

1. Define the variable $\mathbf{z}$ that satisfies Equation (3.1) and draw samples $\mathbf{z}^i$ from $p(\mathbf{z}|y_o)$

2. Based on each sample $\mathbf{z}^i$ drawn in the first step, draw the corresponding samples from $p(\mathbf{x}|\mathbf{z}^i)$

Note that the *original* TSEF can be derived from the *generalized* TSEF as follows:

1. Set $z = H(\mathbf{x}) \in \mathbb{R}$. Use any EDA methods to draw posterior samples $H(\mathbf{x})^i_{posterior}$ from the *univariate* conditional pdf $p(H(\mathbf{x})|y_o)$.

2. Draw from the following Gaussian conditional pdf:

$$p(\mathbf{x}|H(\mathbf{x})^i_{posterior}) = N(\boldsymbol{\beta}_0 + \boldsymbol{\beta}_1 H(\mathbf{x})^i_{posterior}, \boldsymbol{\Sigma}) \tag{3.6}$$

where the parameters $\boldsymbol{\beta}_0$, $\boldsymbol{\beta}_1$ and $\boldsymbol{\Sigma}$ are estimated by the ordinary least squares from prior samples.

It is shown in Grooms (2022) that performing simple linear regressions to update the state variables Δx from the observation increment $\Delta H(x)$ is a deterministic and efficient way to draw sample from $p(\mathbf{x}|H(\mathbf{x})^i_{posterior})$. Note that the first step in the *original* TSEF is a *scalar* problem in the observation space, while the first step can be generalized to a *multivariate* problem in the *generalized* TSEF.

3.2.3 The PFF-DART algorithm

In order to efficiently implement the PFF and facilitate the capabilities in DART, it is beneficial to devise a PFF algorithm that is compatible with the TSEF algorithm in DART. However, it turns out that it is difficult to develop a TSEF algorithm of the PFF that is mathematically equivalent to the original PFF, mainly owing to the presence of the kernel. Therefore, in this section we propose a revised PFF algorithm (PFF-DART) based on the *generalized* TSEF algorithm described in Section 3.2.2. The philosophy of the PFF-DART algorithm is to (1) independently update each state variable (as discussed before) so that the algorithm is parallelizable (2) retain the robust ability of PFF to assimilate non-linear observations.

Intuitively, one can follow the *original* TSEF by using PFF as the EDA method to calculate $\Delta H(x)$, and then perform linear regression to obtain Δx. This should work fine if the observation operator $H(x)$ is linear. However, the linear regression in the second step can be problematic if the observation operator is non-linear. To illustrate this issue, Figure 3.1 shows an example of how TSEF works in an 1-D case. We start from the blue dot on the x-axis x^i_{prior}, which represents the prior model state, and follow the black dashed arrows in the figure. Next, evaluate the observation operator for the prior samples, i.e., the blue dot on the y-axis $H(x^i_{prior}) = H(x)^i_{prior}$. An arbitrary

EDA method is used (i.e., the first step in the TSEF algorithm) to obtain a sample of $p(H(x)|y_o)$ in the observation space, i.e., the solid green dot on the y-axis $H(x)^i_{posterior}$. Finally, update the state variable based on the simple linear regression indicated by the dotted gray line, to obtain the solid yellow dot on the x-axis $x^i_{posterior}$. Note that the linear regression line between z and x (dotted gray line) is determined from all the prior ensemble members (not shown here). If $z = x$ (Fig. 3.1a), then the regression line will have the same slope as the gradient of $H(x)$ and is uniform across the domain. In this case, no matter how much the update in observation space $\Delta H(x)$ is, the update in the state space Δx will be consistent with the update in the observation space, i.e., $H(x^i_{posterior}) = H(x)^i_{posterior}$. On the other hand, if z has a non-linear relation to x (Fig. 3.1b), the posterior samples in the state space $x^i_{posterior}$ obtained from the linear regression (dashed yellow dot) can be inconsistent with the posterior samples in the observation space $H(x)^i_{posterior}$ (dashed green dot), i.e., $H(x^i_{posterior}) \neq H(x)^i_{posterior}$, especially when the increment is large. In the non-linear case, even if we apply a non-linear and non-Gaussian EDA method (e.g., PFF) in the first step to get reasonable increment in the observation space, the desired observation increment cannot be maintained after we linearly update the model space due to this inconsistency.

One possible solution is to choose $\mathbf{z}$ such that is linearly related to $\mathbf{x}$, so that the linear regression will not introduce inconsistencies between $\mathbf{z}$ and $\mathbf{x}$. A naïve choice is $\mathbf{z} = \mathbf{x}$, however we will lose the TSEF since in this case all the updates only depends on the first step and the second step becomes redundant. The TSEF loses it efficiency for this naïve choice.

Instead, the solution we are looking for in PFF-DART is to choose $\mathbf{z}$ to be the so-called "inner domain" in this study. "Inner-domain" (denoted as $\mathbf{z}_y$) is defined to be the subset of the model variables that have direct impact on the outcome of the observation operator. For most of the observations, the inner domain will be a small subset of the model variables that are close to the observation location. For example, if the observation is the radar reflectivity measured at New York, the liquid water mixing ratio at London will not be included in the inner domain for this observation. Since the inner domain is the set of some model variables (similar to the case in Fig. 3.1a), the issue introduced by linear regression will not exist between $\mathbf{z}$ and $\mathbf{z}_y$. Also, because $\mathbf{z}_y$

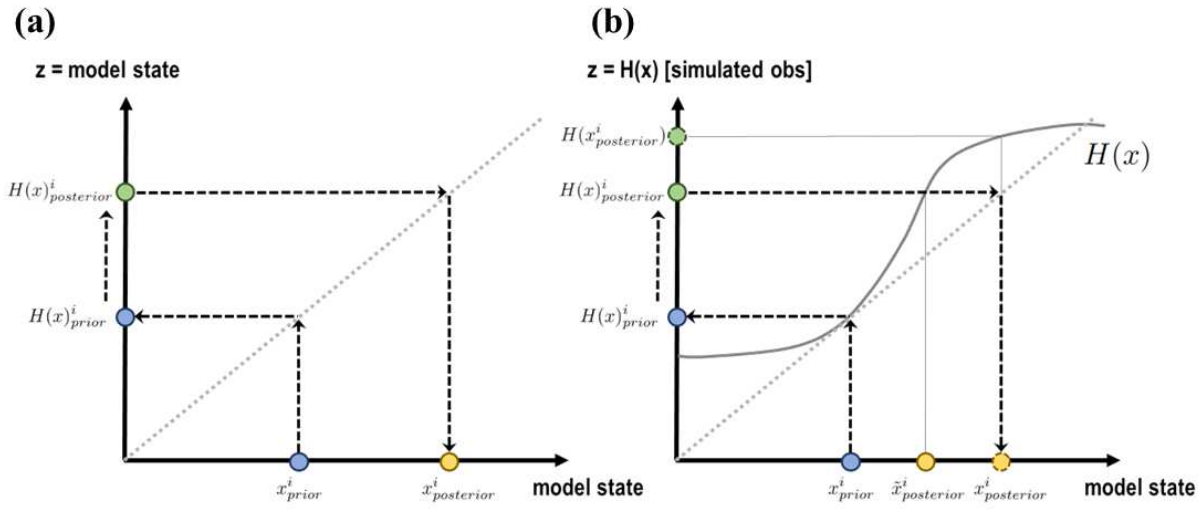

Figure 3.1: Schematic diagram of how the linear regression in the second step of the two-step ensemble filtering (TSEF) algorithm works for (a) linear and (b) non-linear observations. Assume both of x and $z = H(x)$ are scalar here. The dashed black arrow shows the order of how each dot is obtained. The dotted gray line is the linear regression line estimated from the prior ensemble members, which should be an approximation of the local slope of $H(x)$ in this case. In (a): the blue dots are a prior sample in the state space x^i_{prior} (in the x-axis) and in the observation space $H(x)^i_{prior}$ (in the y-axis), the solid green dot $H(x)^i_{posterior}$ is a posterior sample in the observation space that is drawn from $p(H(x)|y_o)$ by any arbitrary Bayesian method in the first step, and the solid yellow dot $x^i_{posterior}$ is the posterior sample in the state space. On the other hand, in (b): the blue dots and the solid green dot are the same as in (a). The dashed yellow dot is the posterior sample in the state space that is drawn from $p(x|H(x)^i_{posterior})$ based on the simple linear regression, while the solid yellow dot $\tilde{x}^i_{posterior}$ is the model state that satisfies $H(\tilde{x}^i_{posterior}) = H(x)^i_{posterior}$. The difference between the solid and dashed yellow dot is due to the non-linearity, which does not exist for linear H and therefore the solid and dashed dots are overlapping in (a). The dashed green dot is the simulated observation from the posterior sample in the state space drawn from the second step by linear regression $H(x^i_{posterior})$, and similarly $H(x^i_{posterior}) \neq H(x)^i_{posterior}$ due to the non-linearity. The gray solid curve in (b) shows the non-linear observation operator.

determines the simulated observations by definition, the inconsistencies between the model state and the simulated observations will no longer exist if we use a proper EDA method to sample $p(\mathbf{z}_y|y_o)$. In addition, since $H(\mathbf{x}) = H(\mathbf{z}_y)$, $\mathbf{z}_y$ satisfies the criterion in Equation (3.1), i.e.,

$$p(y_o|\mathbf{x}, \mathbf{z}_y) = p(y_o|\mathbf{z}_y) \tag{3.7}$$

In other words, $\mathbf{z}_y$ is a valid choice for $\mathbf{z}$ in the *generalized* TSEF algorithm.

The proposed PFF-DART algorithm is summarized as follows:

1. Choose $\mathbf{z} = \mathbf{z}_y$, i.e., the inner domain of y and use PFF as the EDA method to draw samples from a *multivariate* conditional pdf $p(\mathbf{z}_y|y_o)$

2. Draw from the following Gaussian conditional pdf:

$$p(\mathbf{x}|\mathbf{z}_{y}{}^{i}_{posterior}) = N(\boldsymbol{\beta}_0 + \boldsymbol{\mathcal{B}}\mathbf{z}_{y}{}^{i}_{posterior}, \boldsymbol{\Sigma}) \tag{3.8}$$

   where the parameters $\boldsymbol{\beta}_0$, $\boldsymbol{\mathcal{B}}$ and $\boldsymbol{\Sigma}$ are estimated by the multiple linear regression from prior samples.

Similarly, it can be shown that performing multiple linear regression to update the model space from the increments in the inner domain is a deterministic and efficient way to draw sample from $p(\mathbf{x}|\mathbf{z}_{y}{}^{i}_{posterior})$. Based on the above algorithm, once we determine the update of inner domain variables by PFF in the first step, we can independently update each model variable via multiple linear regression, with the predictors being the increment of the inner domain variables. Consequently, PFF-DART can run in parallel within the DART framework.

It is worth noting that simply choosing $\mathbf{z}$ as the inner domain variables $\mathbf{z}_y$ does not solve the issue arising from non-linear observations if we use linear and Gaussian EDA methods in the first step. The essence of PFF-DART is that PFF has the ability to better sample a high-dimensional *multivariate* posterior pdf of $\mathbf{z}_y$ than the other linear and Gaussian methods when the observation operator is non-linear (Hu and van Leeuwen, 2021). In addition, Hu and van Leeuwen (2021) has shown that with the proper choice of the kernel, PFF works well for assimilating non-linear observations in a 1000 dimensional system. The dimension of the inner domain for common geophysical observations is typically less than 100. Therefore, PFF should work stable in the first step of the TSEF algorithm.

Like every other ensemble DA methods, the PFF also suffers from sampling error when the ensemble size is limited. Localization and inflation are common strategies to address the sampling errors. The PFF also needs localization. The PFF-DART algorithm adopts the same increment localization (Anderson, 2012) in DART where the increment of each state variable from an ob-

servation is reduced by a factor that depends on the distance between the observation and the state variable. Here the Gaspari-Cohn function (Gaspari and Cohn, 1999) is used to determine the reduction factor. Although PFF-DART can also adopt other localization strategies like the "B-localization", we hope to focus on the PFF-DART algorithm itself in this study so we stick with the DART-default increment localization method and leave the exploration of other localization methods as future work. On the other hand, numerous inflation methods in DART are also compatible with the PFF-DART algorithm. However, we do not apply any inflation for PFF-DART in this work. Nevertheless, we note that the choice of kernel width (Hu and van Leeuwen, 2021) and the number of iterations (see Section 3.4) in the PFF has a similar effect as the posterior inflation. We will discuss their similarity in Section 3.4 and Appendix B.1.

3.3 Observing System Simulation Experiment (OSSE) Setup

We conduct an Observing System Simulation Experiment (OSSE) to test the PFF-DART algorithm with an idealized atmospheric general circulation model in DART. The aim of conducting an OSSE is to (1) examine the stability of the PFF-DART algorithm (Section 3.4) and (2) evaluate if PFF-DART can improve the assimilation of nonlinear and non-Gaussian observations over linear and Gaussian DA methods (Section 3.5). The OSSE setup and details are described in this section.

3.3.1 Experiment design

In the OSSE, we use a model (the $bgrid\_solo$ model in DART, "Bgrid model" hereafter) which has a modified dynamical core used in the Geophysical Fluid Dynamics Laboratory (GFDL) AM2 atmospheric general circulation model (Anderson et al., 2004). There are four prognostic variables: horizontal wind (U and V), temperature (T) and sea level pressure (slp). The model uses the Arakawa B-grid (Arakawa and Lamb, 1977), and there are 60×30 horizontal grid cells (the horizontal resolution is 6 degrees in both latitude and longitude) and 5 vertical levels. The dimension of the state vector in the Bgrid model is 28,200. With this resolution, the model is capable

of producing baroclinic Rossby waves, which partly resembles the current weather and climate regimes.

We conduct a perfect model OSSE in this study, in which we use exactly the same Bgrid model (with the same model parameters) to generate a truth run and all the other simulations. We generate ensemble members by adding small uncorrelated Gaussian perturbations (in the DART namelist: $perturbation\_amplitude = 0.08$) to a truth model state. The ensemble is then integrated forward in time to the point when the ensemble members reach a climatological RMSE (root mean square error), in which the RMSE of domain-averaged slp is about 5 hPa. (The RMSE of slp increases in the first several months, and then reaches a quasi-steady state and fluctuates between 5-7 hPa, e.g., see Figure 3.3) After this point, we start to assimilate observations every day. Since the model integration timestep dt is 1 hr, the observation frequency is 24 time steps of size dt, for 340 days. Each day we generate 300 observations distributed uniformly over the model domain (Figure 3.2), by adding noise to the truth run. We will compare the truth with the ensemble without DA (noDA), and the ensemble with DA using the Ensemble Adjustment Kalman Filter (EAKF, Anderson (2001, 2003)), and using the particle flow filter (PFF-DART), respectively.

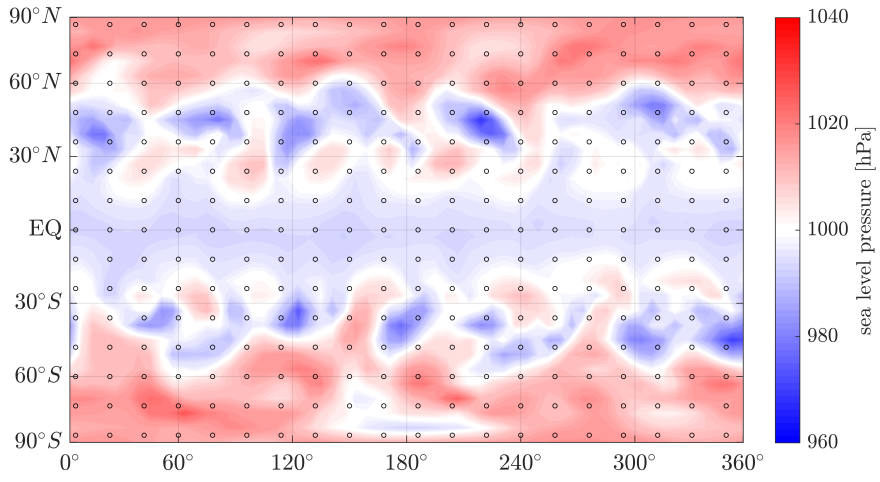

Figure 3.2: The observation network in the Bgrid OSSE experiment. The shading is the sea level pressure of the truth run at the first DA-cycle, and the black circles indicate the observation locations.

3.3.2 Configurations for the filters

The configurations for EAKF and PFF-DART for the default setup are described here. Unless specifically specified, all the experiments follow the default setup in this work. We apply standard DART increment localization for both PFF-DART and EAKF with *cutoff* = 0.2 in the DART namelist. We do not apply any inflation unless specifically mentioned. The ensemble size is $N_p = 25$ for all the experiments.

In the first-step in the PFF-DART (see Equation 2.7), we use a scalar Gaussian kernel $\mathbf{K}$ with a newly developed adapative kernel width algorithm (see Appendix B.1) with $k_{min} = 0.1$, which means the smallest kernel value between all the particles is set to 0.1 in the first iteration. For the gradient of the logarithm of the posterior pdf, we decompose it as

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}|y_o) = \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \nabla_{\mathbf{x}} \log p(y_o|\mathbf{x}) \tag{3.9}$$

where $p(\mathbf{x})$ is the prior pdf and $p(y_o|\mathbf{x})$ is the likelihood. It is challenging to know $\nabla_{\mathbf{x}} \log p(\mathbf{x})$. Although PFF does not require a parametric form of the prior, in practice it is easier to assume a parametric form of the prior and derive its gradient. We assume a Gaussian prior in this work. See Section 3.6 for discussions for potential methods to relax this Gaussian assumption. Therefore, the comparison between EAKF and PFF-DART in this work focuses on the different treatment of the likelihood. $\nabla_{\mathbf{x}} \log p(y_o|\mathbf{x})$ depends on the adjoint of the observation operator $\mathbf{H}^T$ and the gradient of the observation error distribution. Since the observation error in DA has an assigned (usually parametric) form, the gradient of the observation error and the likelihood can be derived. Therefore, we will use the analytical $\nabla_{\mathbf{x}} \log p(y_o|\mathbf{x})$ for PFF-DART. We choose the preconditioner matrix $\mathbf{D}$ as the background error covariance matrix of the inner domain variables. There are a few computational advantages in these choices, which will be detailed in the Appendix B.2. The number of iterations for PFF-DART is set to 30. The learning rate Δs for each iteration is chosen with a revised adaptive learning rate algorithm, with the maximum learning rate Δs_{max} set to 0.1, which is described in Appendix B.3.

3.4 Results for Gaussian likelihood

We first examine the stability and performance of PFF-DART in assimilating observations with a Gaussian likelihood. Specifically, we assimilate linear observations with state-independent Gaussian observation errors. Note that since we assume a Gaussian prior for PFF-DART in this work, a desirable PFF-DART should work similar to a linear and Gaussian filter. Therefore, we use the EAKF ensemble as a benchmark simulation to evaluate the performance of PFF-DART in a year-long cycling DA experiment in this section.

To make the problem slightly more challenging, we *only* generate the slp observations at 300 uniformly distributed locations (Fig.3.2) for each DA cycle. The observation error distribution is $N(0, 1)$ (hPa). Note the the observations are not located on the model grid cells, so the linear slp observation is the linear interpolation of slp at four neighboring model grid points:

$$H(\mathbf{x}) = L(\mathbf{x}) = \sum_{j=1}^{4} w_j x_j \tag{3.10}$$

where w_j is the weight of the j^{th} inner domain that used for the linear interpolation. We set $w_j = \frac{1}{4}$. It is worth mentioning that this linear observation does not have a one-to-one relation to the model variables, so the actual update of the model variables will rely on the statistics in the ensemble.

The performance of the ensemble without DA, with EAKF and with PFF-DART are evaluated based on the domain averaged RMSE and the ensemble spread of U, V, T, and slp, which are shown in Figure 3.3. We can see the strength of DA here: the RMSEs of the ensemble for all the variables with DA (both PFF-DART and EAKF) are much reduced compared to the ensemble without DA, regardless of whether the model variables are observed or not (T, U, V are not observed). The RMSE of EAKF and PFF-DART are very similar for this linear observation. The ensemble spread is similar to the RMSE for EAKF, while the ensemble spread is slightly larger than the RMSE for PFF-DART.

The slightly over-dispersive ensemble in PFF-DART is partly due to the number of iterations we use. We conduct a sensitivity experiment for PFF-DART with different number of iterations,

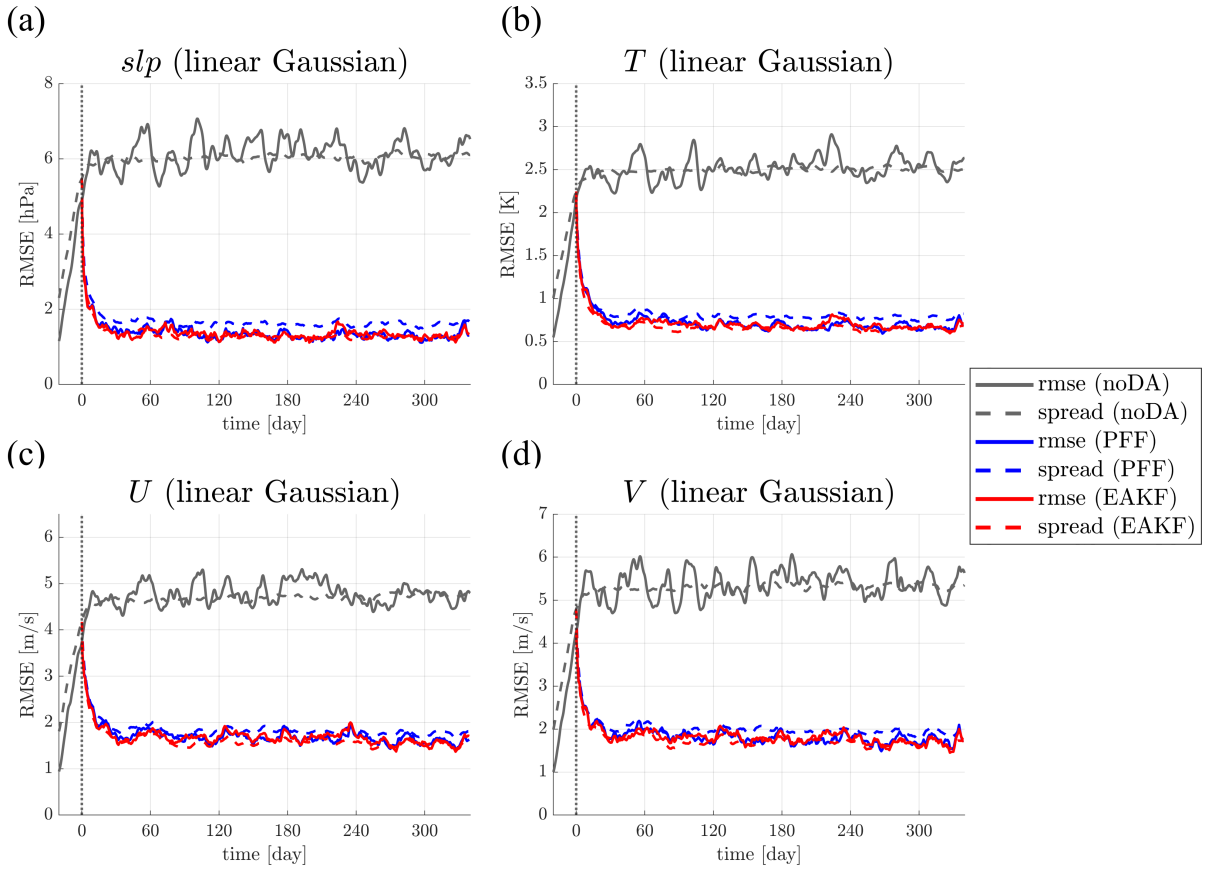

Figure 3.3: The domain averaged root mean square error (RMSE, solid) and the standard deviation (spread, dashed) of the ensemble of noDA (gray), with PFF (blue) and with EAKF (red) experiments for (a) sea level pressure slp (b) temperature T (c) zonal wind U (d) meridional wind V.

and calculate the time averaged RMSE and the ensemble spread in Figure 3.4. We separate the evaluation into two periods: (1) the first 30 days (based on the time evolution plot in Figure 3.3) of the simulation, which is the spin-up period of the DA experiment (Figure 3.4a), and (2) the last 310 days, for which the RMSE and spread have reached a quasi-steady state (Figure 3.4b). During the spin-up period, Figure 3.4a shows that all the PFF-DART simulations and the EAKF have similar average RMSE. The ensemble spread of PFF-DART decreases as more iterations are used, and converges around 80-100 iterations. However, we note that the ensemble spread that PFF-DART converges to is slightly smaller than EAKF and below the 1-1 line, which is an indicator of whether the ensemble and the truth are from the same distribution. This suggests that a fully converged PFF-DART may result in an ensemble that slightly underestimates the ensemble spread. This is similar to the finding in Ba et al. (2021), which suggests that using the deterministic

particle flow could lead to an under-estimated ensemble spread in a high dimension system. On the other hand, during the quasi-steady period, the PFF-DART with 30 and 40 iterations show a similar RMSE to EAKF, while the PFF-DART experiments with more than 60 iterations have a larger RMSE. This can be attributed to the accumulation effect of the too small ensemble spread for the PFF-DART with more than 60 iterations, which gradually under-uses the information in the observations.

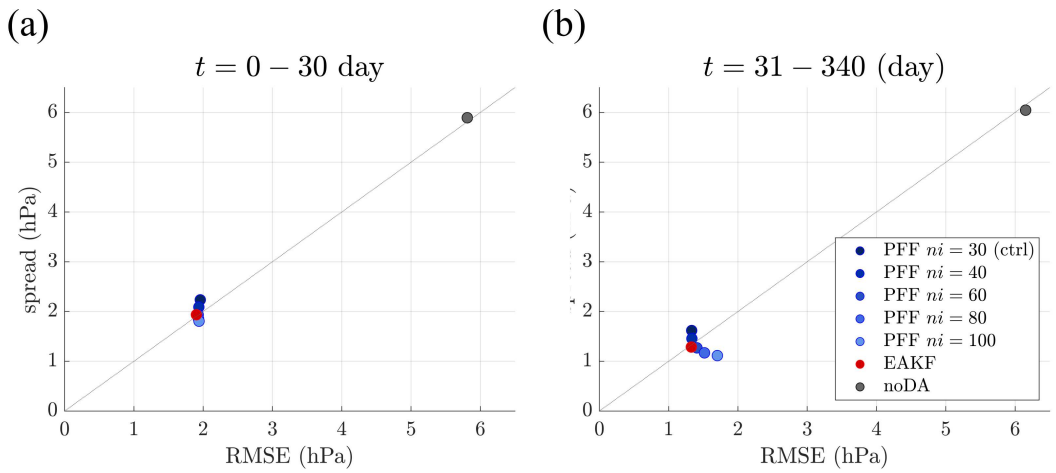

Figure 3.4: The time averaged RMSE and ensemble spread of the slp forecast for different experiments during (a) 0-30 days (the spin-up period) (b) 31-340 days (the quasi-steady period). The blue dots are the PFF-DART simulations with different number of iterations, the red dot is the EAKF simulation, and the gray dot is the noDA simulation. The dashed diagonal line is the 1-1 line, which is an indicator for whether the ensemble and the truth are from the same distribution.

We note that although Figure 3.4 seems to suggest that the optimal number of iterations lies between 40-60, we still stick to the choice of 30 iterations in this work. For one reason, the RMSE is not much affected although the ensemble spread with 30 iterations is slightly too large. In addition, a slightly over-dispersive ensemble is usually preferred over a slightly under-dispersive ensemble. Note that the effect of this "early stopping" strategy is partly similar to posterior inflation. Last but not least, using less iterations makes the algorithm faster, which is always preferred. We note that Leviyev et al. (2022) has proposed a stochastic version of particle flow which alleviates the problem of the under-estimated ensemble spread. We leave the exploration of implementing the stochastic particle flow in DART as a future work.

In addition to the number of iterations, the choice of k_{min} has a similar effect of posterior inflation. We also conduct a sensitivity experiment for the kernel width. Since the results are very similar to the findings in Hu and van Leeuwen (2021), which suggests that using a larger kernel width has the effect of smoothing the posterior pdf, thus resulting in a wider ensemble spread, we leave the results and discussions of the kernel sensitivity experiment in Section B.1.

Overall, albeit slightly over-dispersive, the ensemble from PFF-DART performs very similar to EAKF for the linear and Gaussian observation. This is a good sign since this shows that (1) The PFF-DART algorithm can produce a stable result for this year-long cycling DA experiment (2) It is also able to retain the ability of the linear and Gaussian filter, e.g., EAKF as a benchmark, in assimilating linear and Gaussian observations.

3.5 Results for non-Gaussian likelihood

We next examine the performance of PFF-DART in assimilating the observation with an non-Gaussian likelihood. We examine two types of observations that can make the likelihood non-Gaussian: (1) a linear observation operator but with state-dependent Gaussian observation error (Section 3.5.1) (2) a non-linear observation operator (Section 3.5.2). We note that the occurrence of atmospheric observations with these two types of non-Gaussian likelihood are not uncommon. Examples of both non-linear and state-dependent observations are all-sky satellite radiances (Geer and Bauer, 2011).

3.5.1 Linear observation with a state-dependent Gaussian observation error

We use the same observation operator (i.e., linear interpolation) as in Equation (3.10) in Section 3.4, but generating the observation by adding noises from a state-dependent Gaussian observation error with zero mean and standard deviation which follows:

$$\sigma_o(y_{true}) = 0.1(y_{true} - 970)\ (hPa) \tag{3.11}$$

where y_{true} (hPa) is the true observation, i.e., the observed slp value of the truth at the observation location. The standard deviation of the observation error is smaller (larger) when the true slp at the observation location is lower (higher). Here, we will focus on the comparison between the performance of two experiments: (1) EAKF with a "naïve" Gaussian likelihood $N(y, \sigma_o(y_{true}))$ (2) PFF-DART with the exact non-Gaussian likelihood. We note that since PFF-DART works similar to EAKF for linear and Gaussian observations (Section 3.4), using PFF-DART with the (*incorrectly assigned*) Gaussian likelihood to assimilate non-Gaussian observations works very similar to using EAKF (not shown).

Figure 3.5 shows the RMSE and the spread of the ensemble for different experiments when assimilating linear observations with non-Gaussian likelihood. Compared to Figure 3.3, the average RMSE with DA are slightly larger for this non-Gaussian observation. This is because the average slp is around 1000 (hPa), which corresponds to a larger observation error standard deviation $\sigma_o = 3$ (hPa) from Equation (3.11). Overall, the RMSE of PFF-DART is very similar to EAKF, and the ensemble spread is slightly larger. This is similar to the findings from assimilating linear and Gaussian observations from Section 3.4.

Although the RMSE and the ensemble spread of the forecast in the state space are very similar between PFF-DART and EAKF, the actual ensemble distributions are different. To see the differences, we examine the average analysis mean bias in the observation space, which is

$$b(t) = \frac{1}{300} \sum_{k=1}^{300} \left(\bar{y}_a^{(k)}(t) - y_{true}^{(k)}(t) \right) \tag{3.12}$$

where t stands for the observation time, $\bar{y}_a^{(k)}$ is the analysis (posterior) ensemble mean of the k^{th} observation, and $y_{true}^{(k)}$ is the k^{th} true observation. Figure 3.6 shows the distribution of mean bias for PFF-DART and EAKF. We can clearly see the effect of non-Gaussianity here: the distribution of the biases from the PFF-DART analysis is symmetric around 0, while there are evident negative biases from the EAKF analysis assuming the naïve Gaussian likelihood $N(y, \sigma_o(y_{true}))$.

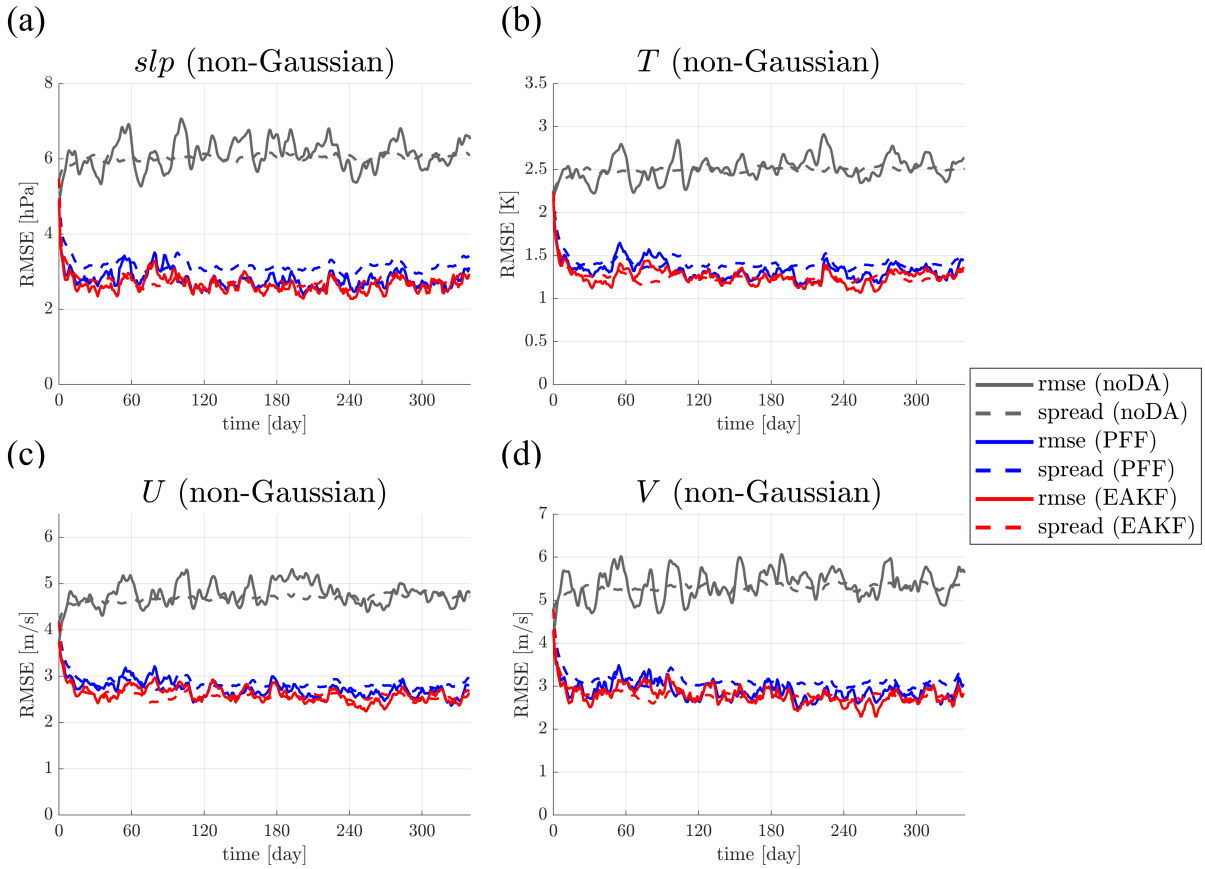

Figure 3.5: Similar to Figure 3.3, but for the non-Gaussian observations.

As a side note, we have also tried another EAKF experiment with Gaussian likelihood but using the ensemble average variance (Bishop, 2019), hereafter, B19-solution. Specifically, the Gaussian likelihood is $N(y, \tilde{\sigma}_o)$ where $\tilde{\sigma}_o$ follows

$$\tilde{\sigma}_o^2 = \frac{1}{N_p} \sum_{i=1}^{N_p} \{0.1(y_i - 970)\}^2 \tag{3.13}$$

and we find EAKF with the B19-solution produces a less asymmetric analysis mean bias distribution (not shown). Note that, however, the experiment setup here is not exactly the same as the assumption from the derivation in Bishop (2019) that the background innovation is unbiased. Although using EAKF with the B19-solution improves the EAKF by reducing the biases in the analysis, assuming a Gaussian likelihood still cannot represent the non-Gaussian posterior pdf.

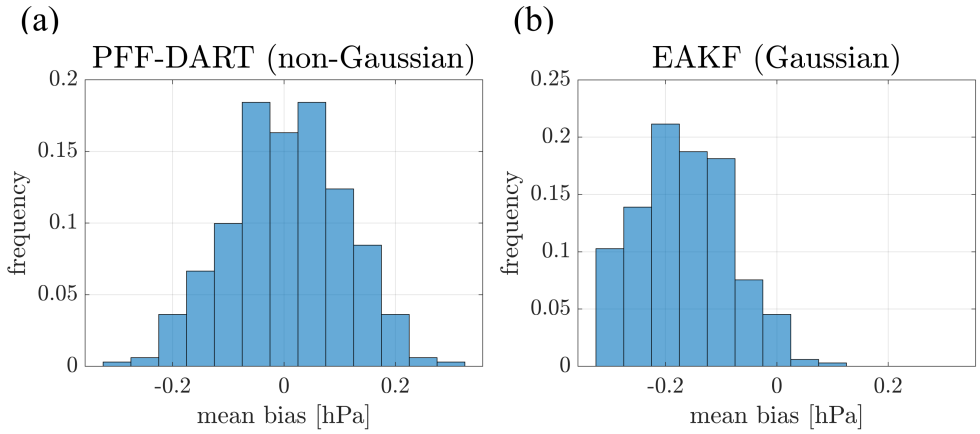

Figure 3.6: The distribution of the average analysis mean bias in the observation space $b(t)$ from (a) PFF-DART (b) EAKF.

To see if the negative biases of the analysis from assuming Gaussian likelihood affect the forecast, we examine the rank histogram of the 1-day forecast of slp in Figure 3.7. The rank histogram of PFF-DART suggests the ensemble forecast is, though slightly over-dispersive, more symmetric (Figure 3.7a) than the rank histogram of EAKF (Figure 3.7b). The rank histogram of EAKF is higher on the high-end and lower on the lower-end, suggesting the ensemble forecast of slp from EAKF also has a negative bias since the truth is more often larger than the whole ensemble.

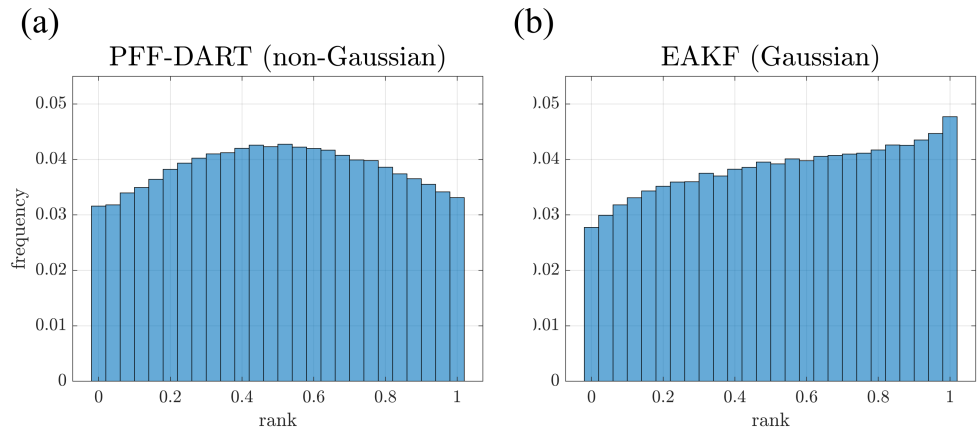

Figure 3.7: The rank histogram of the 1-day forecast of the sea level pressure (slp) from the (a) PFF-DART (b) EAKF experiment when assimilating linear observations with non-Gaussian likelihood.

We remark that although using Gaussian likelihood for the assimilation of non-Gaussian observations in this model does not lead to a significant degradation of the model forecast, it is clear that it still introduces biases in the analysis in the DA cycles. We might just be lucky in this case so that the biases do not further worsen the forecast. In fact, the rank histograms of the 1-day forecast of the unobserved variables are similar between PFF-DART and EAKF (not shown), suggesting the biases in slp do not directly affect the other variables, or the biases are damped by the dynamics of the model in this particular experiment setup.

3.5.2 Non-linear observation

In the second example, we compare PFF-DART and EAKF in assimilating non-linear observations. The non-linear observation operator we use is

$$H(\mathbf{x}) = \exp(L(\mathbf{x}) - p_0) = \exp\left(\sum_{j=1}^{4} w_j x_j - p_0\right) \tag{3.14}$$

where $w_j = \frac{1}{4}$ and $p_0 = 1000$ (hPa). The observation error is assumed to be Gaussian with zero mean and a state-dependent standard deviation as follows:

$$\sigma_o(y_{true}) = \beta y_{true} \tag{3.15}$$

where $\beta = 0.25$. Choosing the standard deviation proportional to y_{true} as in Equation (3.15) for the exponential observation makes the sensitivity of the model variables to the observed value more uniform. We note that if the standard deviation is a fixed number, then a larger observation will result in a much narrower likelihood than a smaller observation for the observation defined in Equation (3.14).

Figure 3.8 shows that PFF-DART can improve the forecast of all the variables for the non-linear observations. On the other hand, the RMSE of EAKF ensemble is reduced during the first 40-50 days compared to the noDA ensemble, while it becomes slightly larger than the noDA ensemble for the rest of the simulation. There are several factors that can explain the failure of EAKF to

improve the forecast, and these factors can mutually influence each other and amplify the negative effect. Observe that in the first 30 days (the spin-up period) of the simulations, the ensemble spread of PFF-DART and EAKF reduces at a similar rate, while the RMSE of EAKF reduces much slower than the PFF-DART. This can be attributed to the incorrect linear regression (Figure 3.1) between the nonlinear observation and the state variables in the second step of the TSEF algorithm for EAKF. After the first 30 days, in addition to the non-linearities, the too small ensemble spread reduces the influence of the observation, and therefore making it difficult for the EAKF to reduce the RMSE. Furthermore, the non-linearity of this exponential observation increases as the RMSE of the background (prior) becomes larger. Therefore, as the RMSE of the EAKF ensemble increases after 30 days, the stronger non-linearity makes it even harder for the EAKF to reduce the RMSE.

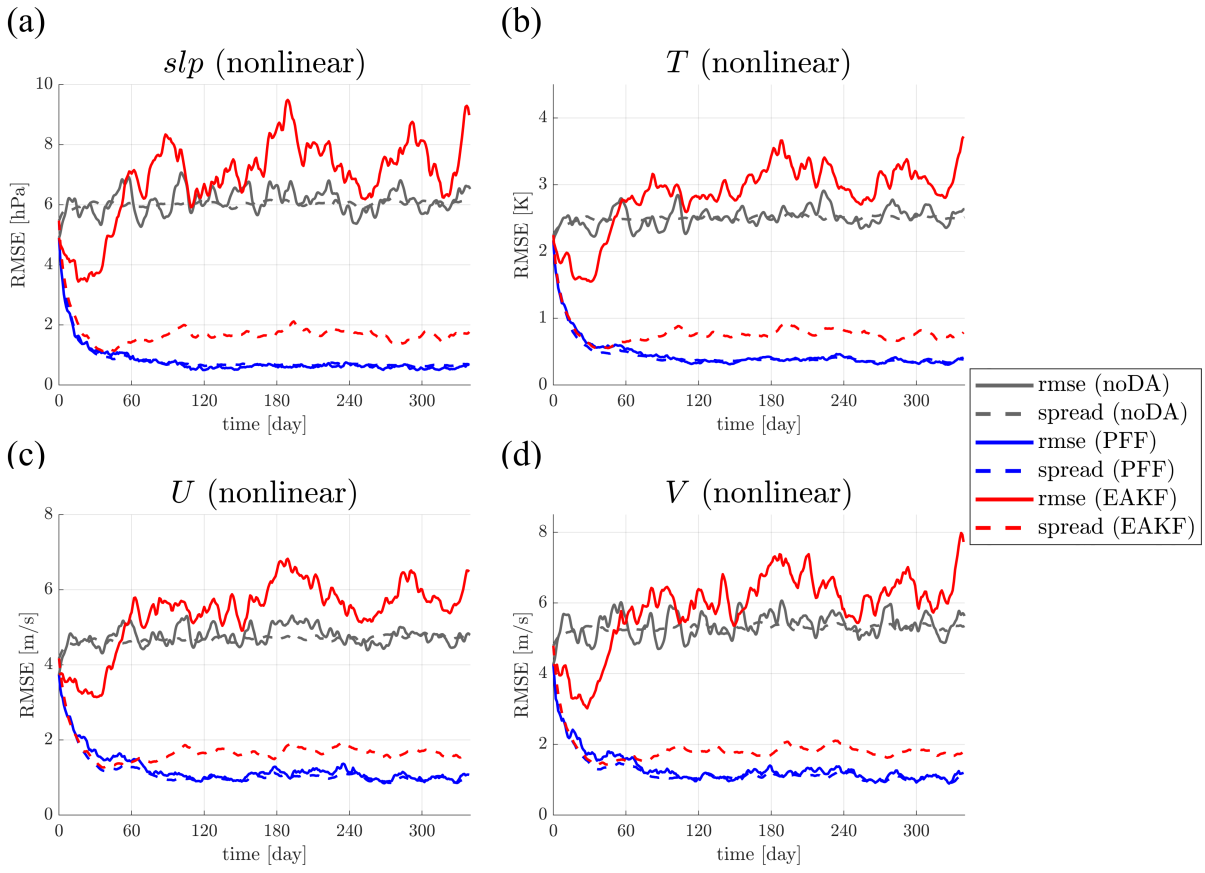

Figure 3.8: Similar to Figure 3.3, but for the non-linear observations.

Since the ensemble spread for EAKF becomes too small during the spin-up period, it would be interesting to examine if some inflation can help EAKF better assimilate the non-linear observation. Here we use the Relaxation-To-Prior-Spread (RTPS) (Whitaker and Hamill, 2012) method to inflate the posterior spread. Specifically, at each grid point, the analysis (posterior) standard deviation of the variable is relaxed toward the background (prior) with a tuning parameter α as follows:

$$\sigma_a \leftarrow (1-\alpha)\sigma_a + \alpha\sigma_b \tag{3.16}$$

where σ_a and σ_b are the analysis and background standard deviation respectively. We conduct a sensitivity test for EAKF with $\alpha = 0.2, 0.4, 0.6, 0.8$. Note that EAKF with RTPS $\alpha = 0$ reduces to the control EAKF. The sensitivity experiment results suggest that the inflation does results in some improvements, especially during the quasi-steady period for $\alpha = 0.4, 0.6, 0.8$ (Figure 3.9b). The reason is that RTPS reduces the decreasing rate of the ensemble spread. Specifically, the slow decreasing rate of RMSE can now catch up with the decreasing rate of the spread so that the filter divergence is reduced. Once the RMSE becomes lower, the non-linearity of this observation also reduces, which makes EAKF easier to assimilate the non-linear observations. With these mutually interacting effects, $\alpha = 0.4$ happens to work best for EAKF in this setup. Nevertheless, we note that the inflation does not really resolve the issues arising from the non-linearities. Specifically, Figure 3.9a shows that the RMSE of all the EAKF experiments with different inflation are similar during the spin-up period, which are all larger than the PFF-DART. This reflects the deficiencies of EAKF in assimilating non-linear observations. The well-tuned EAKF with an optimal inflation still takes much more DA cycles to reduce the RMSE than the PFF-DART, suggesting even with inflation EAKF cannot fully utilize the information in the observation due to the presence of non-linearities.

To demonstrate why the PFF-DART can better assimilate the non-linear observations, we conduct another PFF-DART experiment that assumes that the adjoint of the observation operator $\mathbf{H}^T$ is unavailable. We adopt a similar approach of linear ensemble DA methods to estimate $\mathbf{H}^T$ based

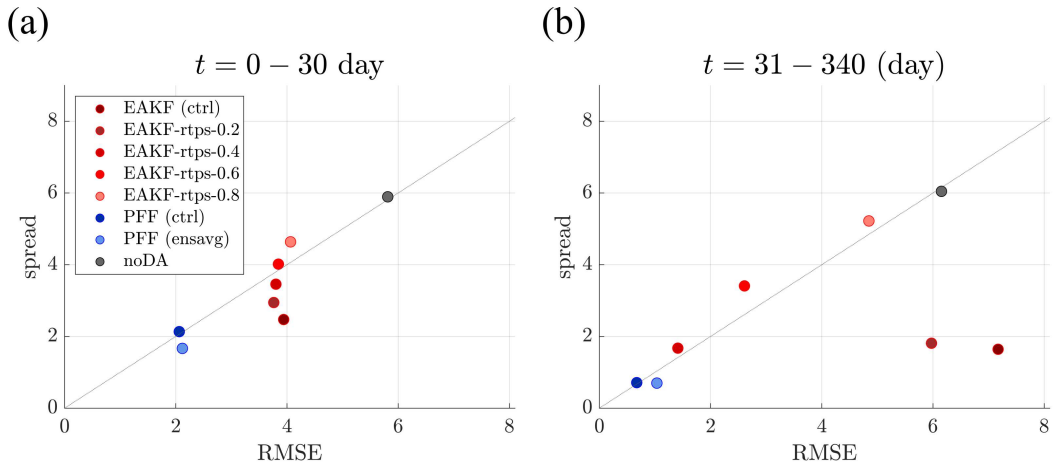

Figure 3.9: Similar to Figure 3.4, but for the sensitivity of EAKF to different values of α in RTPS (red dots). The darker blue dot is the control PFF-DART, and the lighter blue dot is the PFF-DART using the ensemble average $\mathbf{H}^T$.

on linear regression. Specifically, we approximate $\mathbf{H}^T$ based on

$$\mathbf{H}^T \approx \mathbf{Y}\mathbf{Z}_y(\mathbf{Z}_y\mathbf{Z}_y^T)^{-1} = cov(\mathbf{Y}, \mathbf{Z}_y)cov(\mathbf{Z}_y, \mathbf{Z}_y)^{-1} \tag{3.17}$$

where $\mathbf{Z}_y$ is the ensemble perturbation matrix of the inner domain variables and $\mathbf{Y}$ is the ensemble perturbation matrix in the observation space. Note that for the analytical likelihood, $\mathbf{H}^T$ is different for different particles, while here we approximate $\mathbf{H}^T$ for each particle by using the same one in Equation (3.17), which we call the ensemble averaged ("ensavg") gradient. Although we use the same ensavg $\mathbf{H}^T$ for all the particles, we re-evaluate the ensavg $\mathbf{H}^T$ in each iteration using the updated ensemble perturbation matrices $\mathbf{Y}$ and $\mathbf{Z}_y$. In other words, we re-linearize the relation between the inner domain variables and the observation in each iteration. The results show that the PFF-DART with the ensavg $\mathbf{H}^T$ (the lighter blue dots in Figure 3.9) works similar to the default PFF-DART with the exact likelihood (the darker blue dots). Specifically, the RMSE of the ensavg $\mathbf{H}^T$ PFF-DART is much reduced during the spin-up period (Figure 3.9a) compared to EAKFs. This suggests that even without knowing the analytical $\mathbf{H}^T$, we can still use the ensemble statistics in each iteration to better approximate the non-linear relation between the inner domain variables and the observation, thus leading to a better use of the observation information. This quality of PFF

is similar to the outer-loop in the incremental 4D-Var (e.g., Bauer et al., 2010) and the iterative Ensemble Kalman filter (e.g., Evensen, 2018), where the iterative approach in dealing with the nonlinearities can improve the assimilation of non-linear observations.

To summarize, PFF-DART can improve the assimilation of non-linear observations over EAKF. Although in this idealized experiment the non-linearity is much reduced after a number of DA cycles with a well-tuned EAKF with RTPS ($\alpha = 0.4$), we can still envisage in a practical implementation there will still be large innovations even after a long DA cycles, e.g., due to the presence of other error sources that are not properly accounted for. EAKF will still suffer from the issue of non-linearities in these cases, where the PFF-DART, in contrast, has the potential to improve the assimilation over linear DA methods.

3.6 Concluding remarks

In this study, we devise a new algorithm, PFF-DART, in order to efficiently implement the particle flow filter (PFF) in the Data Assimilation Research Testbed (DART). Specifically, each state variable can be updated by each scalar observation independently, which is consistent with the DART infrastructure and therefore PFF-DART can run in parallel. The theoretical foundation of the PFF-DART algorithm can be connected with the Bayes theorem via the generalized Two-Step Ensemble Filtering (TSEF) algorithm proposed in Grooms (2022).

We examine the PFF-DART algorithm in a year-long Observing System Simulation Experiment (OSSE) with a simplified atmospheric general circulation model. We first test the stability of PFF-DART by comparing its performance with a benchmark filter, the Ensemble Adjustment Kalman Filter (EAKF) (Anderson, 2001), in assimilating linear and Gaussian observations. The results show that the PFF-DART can produce a very similar Root Mean Square Error (RMSE) to EAKF, although the ensemble is slightly over-dispersive (Figure 3.3).

Next, we compare the performance between PFF-DART and EAKF in assimilating observations with non-Gaussian likelihood. In the first example, we assimilate linear observation with a state-dependent Gaussian observation error. Although the RMSE of PFF-DART and EAKF are

very similar (Figure 3.5), we find that EAKF introduces biases in the analysis (Figure 3.6). We can also see the biases exist in the 1-day forecast by the rank histogram (Figure 3.7). In contrast, PFF-DART produces a rather unbiased analysis and 1-day forecast, based on a much more symmetric rank histogram. In the second example, we study the behaviors of the two filters by assimilating a non-linear observation. The results suggest PFF-DART can remarkably reduce the RMSE, while EAKF only slightly decreases the RMSE in the beginning of the simulation and then produce similar RMSE to the ensemble without data assimilation (Figure 3.8). Although we can find an EAKF with optimal inflation that can also reduce the RMSE, we note that this well-tuned EAKF needs much more DA cycles to reach a rather low RMSE, which is still larger than the RMSE of the PFF-DART (Figure 3.9). This result suggests that PFF-DART can better utilize the information in the non-linear observations than EAKF (e.g., see Figure 3.1).

One of the innovation in the PFF-DART algorithm and the key to improve the assimilation of non-linear observation is the idea of applying PFF only in the inner domain for each observation, which allows an efficient implementation of PFF in DART and at the same time retaining the important ability of PFF to appropriately assimilate non-linear observations. In addition, with the help of the new efficient algorithm, this is the first time that we demonstrate the feasibility of using PFF to conduct a stable year-long cycling data assimilation experiment in an atmospheric model. Our next step is to examine the performance of PFF-DART in a more sophisticated atmospheric model, e.g., the Weather Research and Forecasting (WRF) model.

With its mixed ensemble and variational nature, along with the the new developments in the TSEF framework, the PFF-DART algorithm has numerous opportunities for integrating exisitng non-Gaussian DA techniques. For example, exploring the non-Gaussian prior for the inner domain, e.g., for water vapor. Given the variational nature of PFF, many non-Gaussian developments in 3(4)D-Vars, e.g., the lognormal (Fletcher and Zupanski, 2006a) or the mixed-Gaussian-lognormal distribution (Fletcher and Zupanski, 2006b; Fletcher, 2010), can be applied in PFF. On the other hand, given the ensemble nature of PFF, we can also explore more flexible distributions such bi-Gaussian (Chan et al., 2020) or Gaussian mixtures (Pulido and van Leeuwen, 2019). Recent

developments for identifying if the prior is Gaussian (Kurosawa and Poterjoy, 2023), or choosing the prior distribution adaptively based on machine learning (Goodliff et al., 2022) can also be used. In addition, recall that PFF does not require a parametric form of the distribution, instead it only requires the estimate of $\nabla_{\mathbf{x}} \log p(\mathbf{x})$. We are now looking into methods that can directly estimate $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ either purely based on the particles, or combined with the Gaussian anamorphosis or the integral probability transform techniques. Furthermore, in the current PFF-DART algorithm, the multiple linear regression in the second-step implicitly assumes a Gaussian relation between the inner domain variables and the other state variables. We may also improve the update in the second-step using the integral probability transform methods discussed in Grooms (2022); Anderson (2023).

There are also other settings in PFF-DART algorithm that merit further investigations. For example, as discussed in Section 4, exploring if using the stochastic particle flow (Leviyev et al., 2022), or other matrix-valued kernels (Hu and van Leeuwen, 2021) can mitigate the issue of the underestimated ensemble spread. Besides, finding a cost-effective way to do the iterations may be an important issue for practical implementations. This also suggests that more research into ways to accelerate the PFF iterations is needed. To give an example, we could explore using the proposal density (van Leeuwen, 2010) to draw particles that are closer to the posterior as the first-guess to replace the first several PFF iterations. We could also use a larger kernel width in the first several iterations to push all the particles closer to the mode, and then use small kernel width in the latter iterations to finely adjust the particles positions to represent the posterior uncertainty. Last but not least, we note that in PFF-DART it is not necessary to set the intermediate variable $\mathbf{z}$ as the inner domain $\mathbf{z}_y$. When assimilating an observation with a large inner domain, we might want to explore setting $\mathbf{z}$ as the coefficients of the leading principal components, also known as loadings, of the inner domain. In this case, although Equation (3.1) might not hold exactly, but with enough number of loadings in $\mathbf{z}$, Equation (3.1) should be a fair approximation.

In summary, we have demonstrated that the newly developed PFF-DART algorithm can improve the non-Gaussian and non-linear DA in a high dimensional geophysical setup. Although we

recognize there are some naïve assumptions in the current PFF-DART algorithm, the versatility of the PFF-DART allows it to be combined with many other innovative DA developments, which further enhances the potential of PFF-DART to be implemented in a more sophisticated geophysical DA problem in the future.

Chapter 4

A non-Parametric way to estimate observation errors based on ensemble innovations[2]

In the second part of the dissertation, we turn our attention to $p(\mathbf{y}|\mathbf{x})$. Recall that in Chapter 1 that $p(\mathbf{y}|\mathbf{x})$ is constructed based on the distribution of the observation error ε^o. We will look into how the distribution of ε^o has been estimated, and then propose an innovative approach to generalize exisiting methods to estimate a non-parametric ε^o in this chapter.

4.1 Background and motivation

The observation error not only includes the uncertainty due to the measurement error, but also the uncertainty due to the representation error. The measurement error is directly related to the measurement instrument design, which is typically well-known. The representation error is defined as the misfit between the observation and the simulated observation resulting from different representations of the reality (see e.g., Hodyss and Nichols, 2015; Van Leeuwen, 2015; Janjić et al., 2018) and hence is directly associated with the observation operator that we have access to. However, the representation error is typically not well-understood. We often do not know the true mapping between the state space and the observation space. For example, for radar simulators, we typically assume an empirical drop size distribution (e.g., Bringi and Chandrasekar, 2001), which is not always realistic. Furthermore, observations and models can represent different spatial scales, which is especially problematic if the observations represent smaller scales than the model can simulate. Nevertheless, there are some strategies to estimate the representation errors, for example, by conducting field campaigns and collecting extensive observations to estimate the relationship between observations at different scales, or performing high-resolution model runs to infer the

[2]This chapter has been submitted to *Quaterly Journal of the Royal Meteorological Society* and is currently under review. Minor revisions are made in Section 4.1 to make the dissertation more coherent.

observation operator (e.g., Hodyss and Nichols, 2015; Van Leeuwen, 2015; Janjić et al., 2018). However, these methods are often less practical to estimate the representation error for every kind of the observation operator.

Instead of having to estimate the measurement error and the representation error separately, there are some statistical ways to quantify the observation error in total, by comparing the observation with several other collocated reference datasets. The reference could most typically be another observation type, or fields from a data assimilation system (background or analysis). It is, however, still difficult to estimate the observation error this way since the reference against which we compare the observation will also have errors. The problem is ill-posed if we do not know the errors in the reference.

One solution is the three-corned hat (3CH) method (Grubbs, 1948; Anthes and Rieckh, 2018; Rieckh and Anthes, 2018), which compares the observation to two independent references. 3CH method can estimate the error variances of the three datasets simultaneously, without any prior knowledge of the errors in the reference, and therefore making the problem well-posed. On the other hand, the most typical solution to make the problem well-posed is to first estimate the errors in the reference, and then to subtract them. For example, if the reference is the model background, then its errors are already estimated as part of the data assimilation system. This leads to the "background error subtraction method" (e.g., Andersson, 2003). In addition, Desroziers et al. (2005) showed how the product of the analysis and background departures can form an alternative way of diagnosing the observation error (their Equation 3). This approach has become a popular method for estimating observation errors (e.g., Bormann and Bauer, 2010; Bormann et al., 2016), and is normally referred to as the "Desroziers method". However, both the background error subtraction method and the Desroziers method make linear assumption of the observation operator, which makes them sub-optimal when the observation is nonlinear. One can refer to, e.g., Sjoberg et al. (2021) and Semane et al. (2022) for the comparison between 3CH and the Desroziers method for further details.

Unfortunately, all of the methods mentioned previously can only estimate the second moment of the observation error pdf. This is fine when the true observation error pdf is Gaussian. However, the second moment is not enough to represent the full non-Gaussian observation error pdf. There are many observations that have non-Gaussian observation errors. For example, for observations with complicated representation errors, there is no reason to assume their observation errors are still Gaussian. Furthermore, when the observed value is close to the boundary for bounded observations, e.g., precipitation or the concentration of some variables, the observation error can be skewed, and hence clearly non-Gaussian (e.g., Lien et al., 2013; Bishop, 2016).

Therefore, the goal of this study is to propose a new method, the Deconvolution-based Observation Error Estimation (DOEE) method, that can more accurately estimate the full observation error pdf. Specifically, DOEE is based on the background error subtraction method (see Section 4.2.2 for details), but here we extend it by relaxing two of the assumptions: the linear approximation for the observation operator and the implicit Gaussian assumption for the observation error pdf. By relaxing these two assumptions, we can obtain an estimate for the full observation error pdf. In addition, we will also discuss ways to construct a state-dependent observation error pdf model using DOEE. Note that the focus of this study is on the theoretical aspect and the demonstration of the new method. We will briefly discuss some ways to apply this method for different DA schemes in Section 4.5, while the actual implementation of this new method in a real DA system is left for future work.

The remainder of this chapter is organized as follows. In Section 4.2, we provide the mathematical background for estimating the observation error, and introduce the new framework DOEE. In Section 4.3, we conduct a series of idealized experiments to justify the new method. In Section 4.4, we apply DOEE to construct a state-dependent observation error model for satellite radiances. Finally, in Section 4.5 we conclude the findings in this paper and propose the use of DOEE in several applications.

4.2 Methodology

4.2.1 Definitions and notations

Since the true atmosphere is a continuum, the true atmospheric state lives in a space of infinite dimension. Let's define $\widetilde{X}^{truth}$ as a random variable representing the truth in this infinite dimension space (hereafter, "continuous space"), and define X^{truth} as a random variable representing the discretized truth in the "model space" (or, the "state space" defined in Chapter 1). Let r be the transformation from the continuous space to the model space:

$$X^{truth} = r\left(\widetilde{X}^{truth}\right) \tag{4.1}$$

The "true observation" Y^{truth} can be related to $\widetilde{X}^{truth}$ as follows:

$$Y^{truth} = \widetilde{H}^t\left(\widetilde{X}^{truth}\right) \tag{4.2}$$

where $\widetilde{H}^t$ is the true observation operator that maps from the continuous space to the observation space. Since the measurement noise is unavoidable in every measurement process, the "actual observed value" Y can be written as

$$Y = Y^{truth} + \varepsilon^m = \widetilde{H}^t\left(\widetilde{X}^{truth}\right) + \varepsilon^m \tag{4.3}$$

where ε^m is the measurement error. Denote the observation operator that we have access to as H. We can rewrite Equation (4.3) as

$$Y = H\left(X^{truth}\right) + \left[\widetilde{H}^t\left(\tilde{X}^{truth}\right) - H\left(X^{truth}\right)\right] + \varepsilon^m \tag{4.4}$$

and note the difference between $\widetilde{H}^t$ and H in Equation (4.4): $\widetilde{H}^t$ is the true mapping (which we do not know exactly) from the continuous state to the observation space, while H is the imperfect observation operator (which we have access to) from the model space to the observation space, which

can include some unrealistic assumptions. Based on the likelihood function in Bayes' theorem, the observation error is defined as

$$\varepsilon^o = Y - H\left(X^{truth}\right) \tag{4.5}$$

Based on Equations (4.4) and (4.5), the observation error can also be written as

$$\varepsilon^o = \left[\widetilde{H}^t\left(\widetilde{X}^{truth}\right) - H\left(X^{truth}\right)\right] + \varepsilon^m \tag{4.6}$$

We define the difference between the true observation $Y^{truth} = \widetilde{H}^t\left(\widetilde{X}^{truth}\right)$ and the simulated observation mapped from the truth in the model space to the observation space by the imperfect observation operator $H\left(X^{truth}\right)$, as the representation error ε^r:

$$\varepsilon^r = \left[\widetilde{H}^t\left(\tilde{X}^{truth}\right) - H\left(X^{truth}\right)\right] \tag{4.7}$$

Therefore, with Equations (4.6) and (4.7), we have

$$\varepsilon^o = \varepsilon^r + \varepsilon^m \tag{4.8}$$

4.2.2 The background subtraction method

Based on the discussions in the introduction, estimating the representation error directly can be challenging. However, the background subtraction method can estimate the total observation error based on the innovation statistics. First consider the following random variable, the innovation D as:

$$D = Y - H(X) \tag{4.9}$$

where X represents a random variable for the discretized model state in the model space. We can further write Equation (4.9) as:

$$D = \left[Y - H\left(X^{truth}\right)\right] + \left[H\left(X^{truth}\right) - H(X)\right] = \varepsilon^o + \left[H\left(X^{truth}\right) - H(X)\right] \tag{4.10}$$

Here we have used the definition of the observation error in Equation (4.5).

We can further linearize the observation operator as follows:

$$H\left(X^{truth}\right) - H(X) \approx \mathbf{H}\left(X^{truth} - X\right) \tag{4.11}$$

For clarity, we use X^b instead of X in the following when we only have one model state (instead of having an ensemble of model states). We can rewrite Equation (4.10) as:

$$D = Y - H\left(X^b\right) \approx \varepsilon^o + \mathbf{H}\left(X^{truth} - X^b\right) = \varepsilon^o - \mathbf{H}\varepsilon^b \tag{4.12}$$

where $\varepsilon^b \equiv X^b - X^{truth}$ is the background error. Assume ε^o and ε^b are independent, then we can estimate the second moment of the pdf of ε^o based on

$$E\left[\left(Y - H\left(X^b\right)\right)\left(Y - H\left(X^b\right)\right)^T\right] = \mathbf{R} + \mathbf{HBH}^T \tag{4.13}$$

where $E[\ldots]$ is the expectation operator, $\mathbf{R}$ is the observation error covariance, and $\mathbf{B}$ is the background error covariance. This is the standard background error subtraction method (Andersson, 2003). We can use an ensemble to provide the flow-independent information of $\mathbf{B}$. Specifically, we can use the ensemble mean $\overline{H\left(X\right)}$ to represent X^b in Equation (4.13):

$$E\left[(Y - \overline{H(X)})(Y - \overline{H(X)})^T\right] = \mathbf{R} + \mathbf{HBH}^T \tag{4.14}$$

and use the ensemble covariance $\mathbf{B}_e$ to approximate $\mathbf{B}$ in Equation (4.14). This method has been used to estimate the observation error variance and correlation (e.g., Bormann and Bauer, 2010; Bormann et al., 2016; Bonavita et al., 2020). Although not stated explicitly, when the ensemble covariance is used to approximate the background error covariance, it is implicitly assumed that the ensemble of model states are independent draws from the same distribution of the true state (i.e., they are independent and identically distributed random variables, aka the i.i.d assumption). See Appendix C.1 for details. Note that this is a common assumption in ensemble DA, and the

basis behind the use of rank histograms (e.g., Hamill, 2001). The i.i.d assumption will also be used for the newly proposed method, which will be detailed later.

As discussed in the introduction, there are two major limitations of the background subtraction method: the linear approximation in Equation (4.11) and the implicit Gaussian assumption for the observation error pdf. First, for nonlinear observations, the linear assumption can only be used when either the background error is small or the observation operator is weakly nonlinear. This assumption might not be true for some observations, e.g., satellite radiances (e.g., Bauer et al., 2010; Bonavita et al., 2018). Second, the second moment is not enough to provide the full observation error when the observation error is non-Gaussian (e.g., Pires et al., 2010).

4.2.3 Deconvolution-based Observation Error Estimation (DOEE) method

In the following, we propose a new method that can estimate the full non-parametric pdf of the observation error by relaxing the two assumptions. We use a general result from statistics that the pdf of a sum of two independent variables is the convolution of the pdfs of those two variables. Therefore, based on Equation (4.10), we have

$$f_D(d) = \int_{-\infty}^{\infty} f_{H\left(x^{truth}\right)-H(X)}\left(d-\hat{\varepsilon}^o\right) f_{\varepsilon^o}\left(\hat{\varepsilon}^o\right) d\hat{\varepsilon}^o \tag{4.15}$$

where f_D, $f_{H\left(X^{truth}\right)-H(X)}$, and f_{ε^o} is the pdf of the random variable D, $H\left(X^{truth}\right) - H\left(X\right)$, and ε^o respectively. Note that the assumption that ε^o and $\left[H\left(X^{truth}\right) - H\left(X\right)\right]$ are independent is equivalent to assuming that the observation error and the background error are independent. If the pdf $f_{H\left(X^{truth}\right)-H(X)}$ is known we can use statistics on the innovations to estimate f_{ε^o} by deconvolution. This is the essence of our method, and therefore we call this the "Deconvolution-based Observation Error Estimation (DOEE)" method. Note that Equation (4.15) is completely general: we neither require a linear (or weakly nonlinear) observation operator, nor a Gaussian (or any distribution) distributed model or truth state. Furthermore, DOEE can provide us the full pdf, instead of just the second moment of the observation error pdf. We will discuss a practical method

to find the first two pdfs in Sections 4.2.4 and how to solve Equation (4.15) in Section 4.2.5 and Appendix C.2.

It is interesting to understand the meaning behind Equation (4.15). The convolution relationship states that f_D is a smoothed version of f_{ε^o}, in which the smoothing kernel is $f_{H\left(X^{truth}\right)-H(X)}$. The smoothness depends on the width and the shape of $f_{H\left(X^{truth}\right)-H(X)}$. In general, the wider $f_{H\left(X^{truth}\right)-H(X)}$, the stronger the smoothing effect is on f_D. Similarly, for the same innovation pdf f_D, if the pdf $f_{H\left(X^{truth}\right)-H(X)}$ (i.e., the smoothing kernel) is narrower, we can expect a wider observation pdf f_{ε^o}.

4.2.4 A practical method to estimate the pdfs

It is rather straightforward to estimate f_D. Specifically, we can use the histogram of the differences between the observation and the collocated ensemble of model equivalences. However, estimating $f_{H\left(X^{truth}\right)-H(X)}$ requires the knowledge of $f_{H\left(X^{truth}\right)}$, in which we do not know the truth exactly. Nevertheless, we do not actually need to know the truth, instead, we only need its distribution. Therefore, a practical way to estimate $f_{H\left(X^{truth}\right)-H(X)}$ without the need to know the truth, is based on an ensemble of model states. In particular, similar to the implicit assumption that uses the ensemble covariance $\mathbf{B}_e$ to approximate the true background error covariance $\mathbf{B}$, we assume the ensemble members and the true state are independent draws from the same distribution (the i.i.d asumption). In other words, the pdf of X^{truth} is the same as the pdf of X. Therefore, the pdf of $H\left(X^{truth}\right)$ is also the same as the pdf of $H\left(X\right)$. With this i.i.d assumption, we can randomly pick one ensemble member in the observation space and treat it as a draw for $H\left(X^{truth}\right)$. Its differences from the other ensemble members in the observation space then provide us independent samples of $f_{H\left(X^{truth}\right)-H(X)}$.

The above estimation of $f_{H\left(X^{truth}\right)-H(X)}$ could be sensitive to which member we pick as the truth member when the sample size is small. However, we find that when the sample size is large enough, the result does not change much as we pick a different member (we will verify this in the sensitivity experiment in Section 4.3.2). When the sample size is small, we can mitigate the

sampling error by averaging the histograms obtained from picking each member as the truth. This is equivalent to using the pairwise differences of the ensemble members as samples drawn from $f_{H\left(X^{truth}\right)-H(X)}$.

We can also estimate state-dependent observation errors. Suppose the observation error pdf depends on a predictor C, which is a function of the true state, $C = C\left(X^{truth}\right)$. We can rewrite Equation (4.10) conditioned on C as

$$D_{|C\left(X^{truth}\right)} = Y_{|C\left(X^{truth}\right)} - H(X) = \varepsilon^{o}_{|C\left(X^{truth}\right)} + \left[H\left({X^{truth}}_{|C\left(X^{truth}\right)}\right) - H(X)\right] \tag{4.16}$$

Note that we have used the fact that X^{truth} and X are i.i.d, so $X_{|C\left(X^{truth}\right)} = X$. Since the pdf of $H\left({X^{truth}}_{|C\left(X^{truth}\right)}\right)$ is equal to the pdf of $H\left(X_{|C(X)}\right)$, we draw from a subset of ensemble representing $H\left(X_{|C(X)}\right)$ to sample $H\left({X^{truth}}_{|C\left(X^{truth}\right)}\right)$ instead of arbitrarily picking one from the whole ensemble. We call this subset of ensemble representing $H\left(X_{|C(X)}\right)$ the "indistinguishable subset". Similarly, when the sample size is small, we can average the histograms of the differences between each member in the indistinguishable subset and the other ensemble members to approximate the pdf of $H\left({X^{truth}}_{|C\left(X^{truth}\right)}\right) - H\left(X\right)$. Note that in real cases, we need an approximation for $C\left(X^{truth}\right)$, e.g., using the observations such that $C\left(X^{truth}\right) = C_{obs}$, since we do not know the truth.

4.2.5 The way to solve the deconvolution equation in DOEE

In this section, we present some of the mathematical details of how we solve the deconvolution equation in Equation (4.15) in DOEE. We express the pdfs f_D and f_{ε^o} using histogram with bins centered at $[-n,\ -n+1, \ldots, n]\,\Delta x$, where Δx is the bin-width. We choose Δx based on the Freedman-Diaconis rule (Freedman and Diaconis, 1981), which is designed to select the bin-width that minimizes the squared difference between the histogram generated by the samples and the underlying true pdf. Specifically, the bin-width Δx is chosen based on

$$\Delta x = 2\frac{IQR}{\sqrt[3]{N}} \tag{4.17}$$

where IQR is the interquartile range of the data, and N is the sample size. Then, we choose n so that $n\Delta x$ is large enough and we can make sure the values of pdf close to the boundaries are all zeros. If we discretize Equation (4.15) using histograms defined by the bins above, the value of pdf f_D at $m\Delta x$ can be written as:

$$f_D(m\Delta x) = \sum_{j=-n}^{n} f_{H\left(X^{truth}\right)-H(X)}((m-j)\Delta x) f_{\varepsilon^o}(j\Delta x) \tag{4.18}$$

Note that we have totally $2n+1$ equations (with $m = -n, \ldots, n$ in Equation (4.18)) and $2n+1$ unknowns ($f_{\varepsilon^o}(j\Delta x), j = -n, \ldots, n$), and therefore we can form a linear system of equations. For convenience, we can write the linear system of equations into a matrix form:

$$\mathbf{f}_D = \mathbf{A}\mathbf{f}_{\varepsilon^o} \tag{4.19}$$

where $\mathbf{f}_D$ and $\mathbf{f}_{\varepsilon^o}$ are column vectors with the values from $f_D(j\Delta x)$ and $f_{\varepsilon^o}(j\Delta x)$, $j = -n, \ldots, n$ respectively, and $\mathbf{A}$ is a matrix containing the values of $f_{H\left(X^{truth}\right)-H(X)}(j\Delta x)$ that fulfills Equation (4.18).

Although the number of equations and the unknowns is the same, it turns out that directly solving Equation (4.19) by $\mathbf{A}^{-1}\mathbf{f}_D$ does not give us a desirable solution. Specifically, the pdf can have negative values, and the solution tends to be very noisy. Therefore, we try to seek the solution from a constrained minimization problem, in which we formulate a cost-function with a penalty term to make the solution smooth, and also constrain the solution to be non-negative. There is a "smoothness parameter" in the cost-function to control the smoothness of the solution. Please see Appendix C.2 for details of the cost-function and how we determine the smoothness parameter.

Finally, we summarize how DOEE method works in the following:

1. Collect samples of observations and the collocated ensemble of model equivalences in observation space. Define the indistinguishable subset if constructing a state-dependent error model (we will give an example in Section 4.4.1). Calculate the histograms of $\mathbf{f}_D$ and $\mathbf{f}_{H\left(X^{truth}\right)-H(X)}$.

2. Form the cost-function with the optimal smoothness parameter (see Appendix C.2).

3. The observation error pdf $\mathbf{f}_{\varepsilon^o}$ is the solution that minimizes the cost-function with the optimal smoothness parameter.

4.3 Idealized experiments

In this section, we conduct a series of idealized experiments to demonstrate and justify the DOEE method. In Section 4.3.1, we will first show that DOEE not only can retrieve a Gaussian observation error pdf, but also a non-Gaussian one (bi-modal, skewed), which is the strength of DOEE. In Section 4.3.2, we will examine the sensitivity of DOEE to the sample size, the quality of the ensemble, and the member we pick as the truth.

4.3.1 Experiment setup and results

The setup for the idealized experiments is as follows. First, we generate the truth from a given distribution, and then add noise drawn from an (assumed unknown) observation error pdf, which we would like to estimate, to the truth to generate the observation. We generate the ensemble by drawing from the same distribution as we draw the truth. Then, we use the observation and the ensemble as input to the methodology, and try to estimate the unknown observation error pdf. We then compare the estimated observation error pdf with the true observation error pdf to evaluate the performance of DOEE. If not mentioned differently, the number of observations (grid points) is 10,000, and the ensemble size is 100 in the following experiments.

For all the experiments, we generate the truth and the ensemble by drawing from a Gamma distribution, $Gamma\,(2+S_h,\ 2+S_c)$, where S_h and S_c are independent draws from the Uniform distribution $Uniform\,(-0.5, 0.5)$. The reason we choose a Gamma distribution is that it is often used to simulate the distribution of variables in the geophysical models, for example, the precipitation-related variables (e.g., Bishop, 2016). In addition, it is skewed, which can mimic the situation when the observation operator is non-linear, which would transform a symmetric distribution (e.g., Gaussian) into a skewed distribution. Note that we choose the distribution of the truth

differently at different grid points, and this is designed to mimic the realistic situation as well. In addition, the method should be based on the collection of observations whose observation errors are from the same pdf at different grid points.

We first examine cases where the observation error is Gaussian but with non-zero mean, specifically, $Normal\,(2, 2^2)$ or $Normal\,(-2, 2^2)$. These cases are conducted to mimic situations where there is a bias in the prediction of the observation operator. The results show that we can well reproduce the true observation error (Figure 4.1a,b), suggesting this method can also be applied for bias correction.

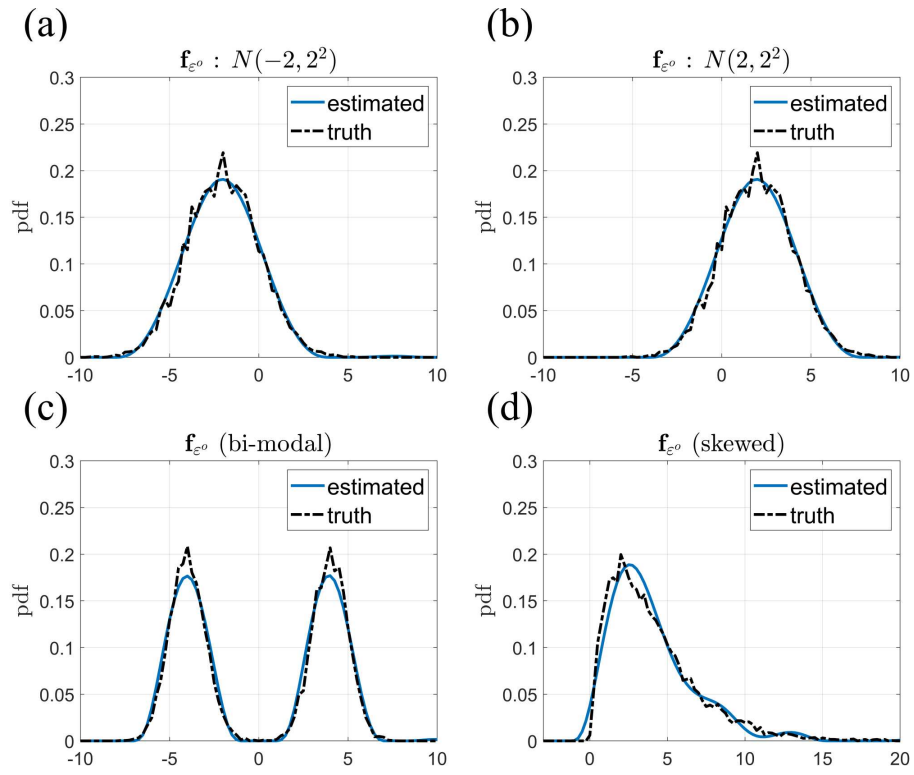

Figure 4.1: The true observation error pdf (black dashed line), and the estimated (blue solid line) observation error pdf from DOEE method. These are the cases where the observation error pdf is (a) Gaussian with a negative mean, (b) Gaussian with a positive mean, (c) bi-modal Gaussian and (d) Gamma.

Next, we turn to some more difficult cases where the observation error is non-Gaussian. Specifically, Figure 4.1c shows the result when the true observation error is drawn from a bi-Gaussian distribution $\frac{1}{2}\left[Normal\,(-4, 1^2) + Normal\,(4, 1^2)\right]$, and Figure 4.1d is when the observation error

is from $Gamma\,(2,2)$. We can see the estimated error pdfs are very close to the true error pdfs. These results are encouraging, since they suggest that DOEE has the ability to well capture non-Gaussianities in the observation error. Especially, we can accurately retrieve the error pdf when the true error is bounded and skewed (Fig.4.1d). For example, for semi-positive definite observations, when the observed value is zero, the realization of the observation error can only be either negative or zero. In this case, the observation error pdf has to be skewed if the error variance does not vanish. Previous methods are not able to provide any information of the skewness for this kind of observation error, while our method can quite accurately capture the full pdf.

We examine all the pdfs from Equation (4.15) in Figure 4.2. As discussed before, the innovation pdf (red curve) is the smoothed observation error pdf (blue curve), where the smoothing kernel is the ensemble difference pdf (orange curve). We also calculate the convoluted pdf (brown dashed curve) from the estimated observation error pdf from DOEE (blue curve) and the ensemble difference pdf (orange curve). The convoluted pdf almost overlaps with the innovation pdf, which suggests that the estimated observation error pdf from DOEE well satisfies Equation (4.15) or (4.19).

4.3.2 Sensitivity tests

Recall that one of the assumptions in the method is that the ensemble is indistinguishable from the truth (the i.i.d assumption). However, in practice, it is difficult to know if this assumption is true or not. Therefore, here we examine how the estimate changes when the i.i.d assumption is not true. First, we examine the case that the ensemble is under-dispersive. Specifically, we generate the truth and the ensemble based on the following distributions:

$$H\left(X^{truth}\right) \sim \mathrm{Gamma}\,(2+S_h, 2+S_c) + \mathrm{Normal}\left(0, 1.5^2\right)$$
$$H(X) \sim \mathrm{Gamma}\,(2+S_h, 2+S_c)$$

The ensemble is generated from the same distribution as the previous sections, while the truth is generated by a wider distribution. We take all the observation errors to be Gaussian in this section.

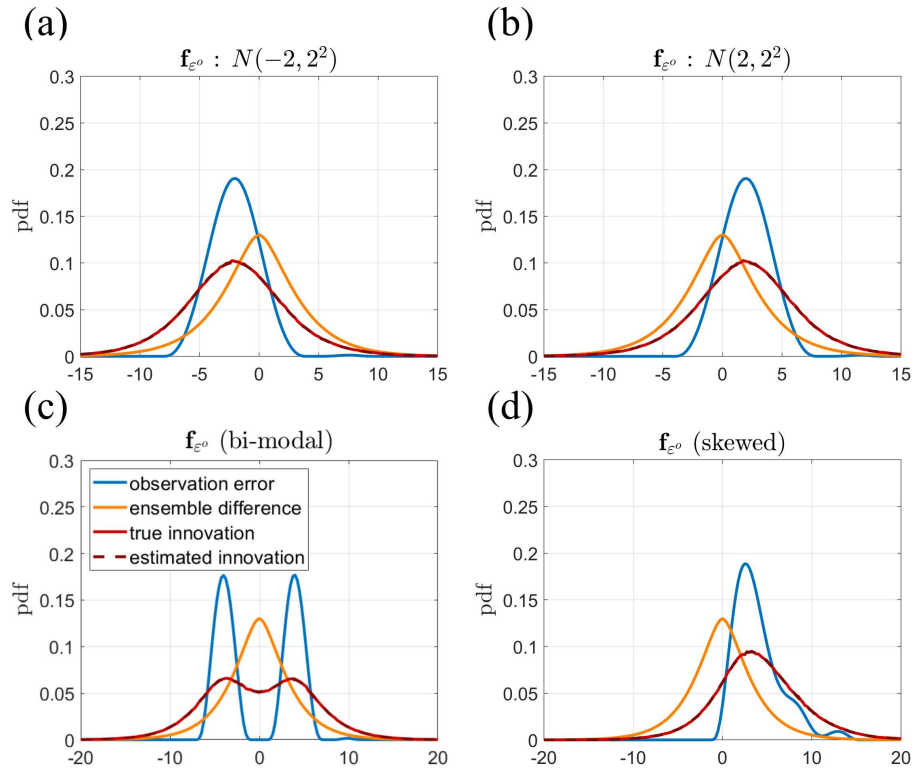

Figure 4.2: The decomposition of the pdfs in Equation (4.15). The blue line is the estimated observation error pdf f_{ε^o} (the same as the blue lines in Fig.4.1), the orange line is the ensemble difference pdf $f_{H\left(X^{truth}\right)-H(X)}$, the solid red line is the innovation pdf f_D, and the dashed brown line is the estimated innovation pdf given by the convolution between the estimated observation error pdf f_{ε^o} and the ensemble difference pdf $f_{H\left(X^{truth}\right)-H(X)}$. These are the cases where the observation error pdf is (a) Gaussian with a negative mean, (b) Gaussian with a positive mean, (c) bi-modal Gaussian and (d) Gamma.

We conduct a similar experiment as described in Section 4.3.1, but in this case the estimation of $f_{H\left(X^{truth}\right)-H(X)}$ is based on the under-dispersive ensemble. Figure 4.3a shows that the estimated observation error becomes over-dispersive. We conduct another experiment when the ensemble is over-dispersive (by swapping the distribution of the truth and the ensemble from the previous experiment), and the estimated observation error becomes under-dispersive (Fig.4.3b).

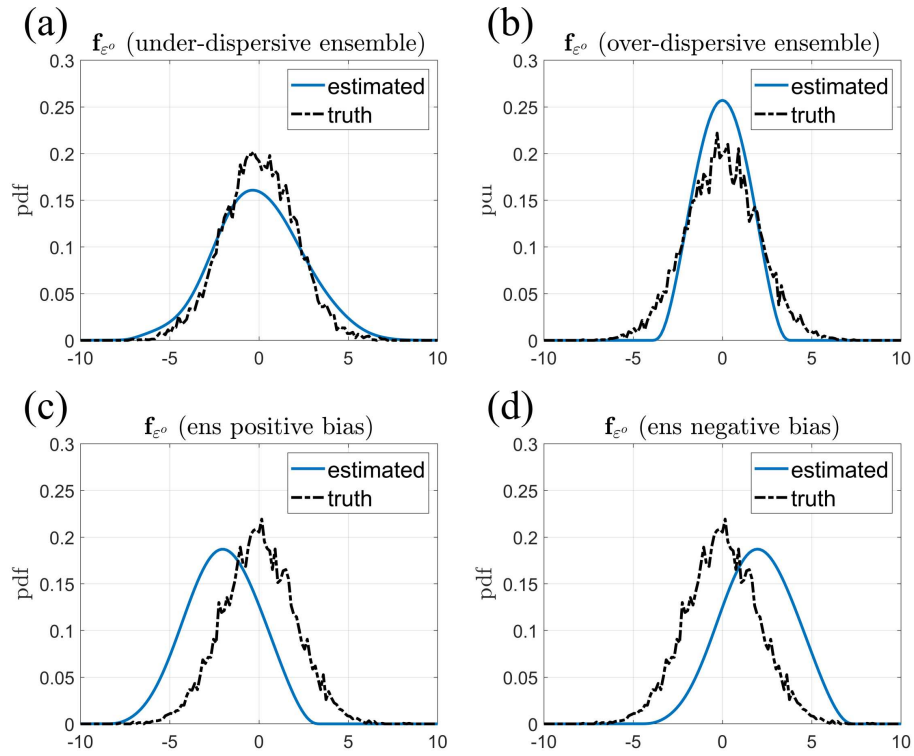

Figure 4.3: Similar to Figure 4.1, but when the i.i.d assumption for the ensemble is invalid. (a) When the ensemble is too narrow, (b) when the ensemble is too wide, (c) when the ensemble has a positive bias, and (d) when the ensemble has a negative bias.

Figure 4.3c-d show the results when the ensemble is biased from the truth state. When the ensemble has a positive bias, we can expect there is a negative bias in the estimated observation error (Fig.4.3c), and vice versa (Fig.4.3d). Recall that in Figures 4.1a-b we have also seen an observation error pdf that is not centered at zero. However, note that in Figures 4.1a-b, the ensemble is drawn from the same distribution as the truth, while that is not the case for Figures 4.3c-d. The biases in Figures 4.1a-b are totally attributed to the errors in the observation operator (or the mea-

surement noise, although less likely), while the biases in Figures 4.3c-d are totally attributed to the poor quality of the ensemble, and hence the errors in the model. In theory, we should deal with these two kinds of biases differently: when the error comes from the observation operator (like in Figs.4.1a-b), we should include them into the observation error. In contrast, if the error comes from the ensemble, we should include them into the background error (like in Figs.4.3c-d), i.e., remove the bias in the prior ensemble model state. In practice, however, since we do not know if the truth observation error is biased or not, we cannot distinguish which is the case when we obtain a biased observation error pdf like in Figures 4.1a-b or 4.3c-d.

Finally, we examine the sensitivity of DOEE to the sample size. Specifically, we are trying to answer the question: how does the choice of the truth member affect the estimation of the ensemble difference pdf $f_{H\left(X^{truth}\right)-H(X)}$ and the observation error pdf f_{ε^o}. Here we examine three ways to estimate the ensemble difference pdf: (1) We always choose the first member as the truth. (2) Every time we randomly pick a different ensemble member as the truth. (3) We calculate the pdf of the pairwise ensemble differences (there is in total $\frac{N_e(N_e-1)}{2}$ number of pairs where N_e is the ensemble size). We compare the true ensemble difference pdf and the estimated one from the three methods with 100 ensemble members and 10,000 observations in Figures 4.4a-c. It is hard to tell the differences between the true pdf and the three estimated pdfs, suggesting that all of the methods provide an accurate estimate of $f_{H\left(X^{truth}\right)-H(X)}$. Since the ensemble difference pdfs are very similar, it is not surprising that the estimated observation error pdfs are also very similar for the three methods (not shown). We redo the experiment, but now reduce the ensemble size from 100 to 10, and the number of observations from 10,000 to 100. The results are shown in Figures 4.4d-f. We do see some minor differences in the fine structure of the estimated ensemble difference pdfs, while the overall shape of the pdfs are still very similar, as are the estimated observation error pdfs (not shown).

We conclude that the estimated observation error pdf is not sensitive to the choice of the truth member, and DOEE works well even for small ensemble sizes and number of observations.

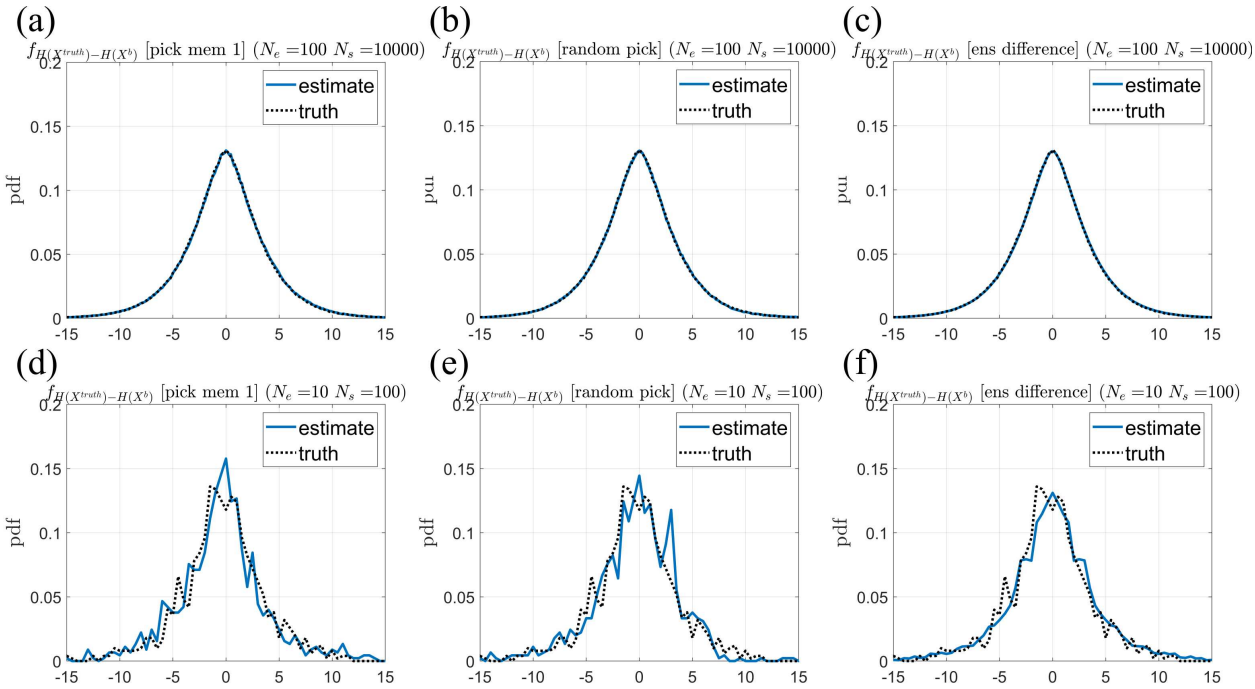

Figure 4.4: The true (black dashed line) and the estimated ensemble difference pdf $f_{H\left(X^{truth}\right)-H(X)}$ (blue solid line). For the estimate, (a)(d) the first member is always chosen as the truth (b)(e) the truth is randomly chosen from the ensemble members (c)(f) the ensemble difference is used for the estimation. The ensemble size N_e and the number of observation N_s are (a)(b)(c) N_e=100, N_s=10,000 and (d)(e)(f) N_e=10, N_s=100.

4.4 An application of DOEE to real data

4.4.1 Background and data descriptions

In this section, we apply DOEE to a real dataset. We will look at the observation at 37 GHz vertical polarization measured by the Special Sensor Microwave Imager/Sounder (SSMIS) (Kunkee et al., 2008). SSMIS is onboard the DMSP (Defense Meteorological Satellite Program) satellite, which is polar-orbiting and provides global coverage about twice per day. We collect the observations during September 2015. The observations are paired with an ensemble, which is from the local ensemble transform Kalman Filter (LETKF) from the European Centre for Medium-Range Weather Forecasts (ECMWF) (Bonavita et al., 2020), and the ensemble size is 100. Due to the limitation of the radiative transfer model, observations over land and over the regions where the latitude is over 60 degrees are excluded in the analyses. After data screening, there are in total around 900,000 pixels, which should provide enough samples for the analysis.

Previous studies have suggested that the observation error for the all-sky radiance observations can depend on the cloud amount (e.g., Geer and Bauer, 2011). Specifically, we assume that f_{ε^o} depends on the cloud amount C, which in this study is defined as,

$$C = 1 - \frac{T_{B;19V} - T_{B;19H}}{T_{B,clear;19V} - T_{B,clear;19H}} \in [0, 1] \tag{4.20}$$

where $T_{B;19V}$, $T_{B;19H}$ are the vertical and horizontal polarized brightness temperature at 19 GHz, and $T_{B,clear;19V}$, $T_{B,clear;19H}$ are the vertical and horizontal polarized brightness temperature at 19 GHz assuming the cloud is absent. The advantage of using Equation (4.20) is that we can have a consistent definition for the cloud amount among the ensemble members and the observations. $T_{B,clear;19V}$, $T_{B,clear;19H}$ are estimated using a high-resolution control run and they are assumed to be the same among the observation and all the ensemble members at the same grid point. Although $T_{B,clear;19V}$ and $T_{B,clear;19H}$ depend on variations in temperature, specific humidity and skin temperature, which can be different for each ensemble member and the control run, the ensemble spread in these variables only cause minor variations in simulated brightness temperature across

the ensemble, when compared to the variations caused by the cloud amount. This supports our approximations for $T_{B,clear;19V}$, $T_{B,clear;19H}$ using the control run. However, since the approximation is still not perfect, especially for the clear-sky end of the cloud amount predictor (e.g., Lonitz and Geer, 2020), sometimes it could lead to a negative cloud amount. Therefore, here we impose a lower bound of zero on the calculated cloud amount. Note that the 19 GHz cloud predictor used here differs from the 37 GHz predictor used in e.g. Geer and Bauer (2011). This is because the 37 GHz h-polarized channel is not actively assimilated and prior brightness temperatures for this channel are not available in the LETKF.

Since the cloud amount is a continuous variable, we discretize the cloud amount when building the cloud amount dependent observation error pdf. In practice, we assume that the observation error pdf f_{ε^o} is the same pdf within a finite range of cloud amount. Recall Equation (4.16) provides us the way to estimate this cloud amount dependent error model. For example, assume we would like to estimate the observation error when cloud amount is between [0.2, 0.3]. We collect the observations with observed cloud amount $C_{obs} \in$ [0.2, 0.3] and calculate the difference between the observations and all of the ensemble members (regardless of their cloud amount), to estimate the pdf of $Y_{|C\left(X^{truth}\right)\in[0.2,0.3]} - H(X)$. To estimate the pdf of $H\left(X^{truth}{}_{|C\left(X^{truth}\right)\in[0.2,0.3]}\right) - H(X)$, we pick an ensemble member from the indistinguishable subset, which in this case is the subset of the ensemble with cloud amount within [0.2, 0.3], and calculate the differences between the member in the indistinguishable subset and the other ensemble members. Note that a partly similar idea was used by Chambon et al. (2014) to estimate the biases of all-sky microwave radiances using a sample in which both observations and background had similar cloud amounts.

To have an idea of the relation between the observed cloud amount and the ensemble cloud amount, we first examine the logarithm of their joint distribution in Figure 4.5. It's interesting to see that the joint histogram is not symmetric. Although it seems that the ensemble has too large cloud amount in general (e.g., compare the values along the 1-4 line and the 4-1 line in Fig.4.5), the ensemble has too small cloud amount in the bin with cloud amount < 0.1. In summary, the observed cloud amount can sometimes be very different from the ensemble cloud amount, and the

spread of the ensemble cloud amount can be very large (we can see, for example, by looking at the marginal pdf of the ensemble cloud amount given a value of the observed cloud amount in Fig.4.5a).

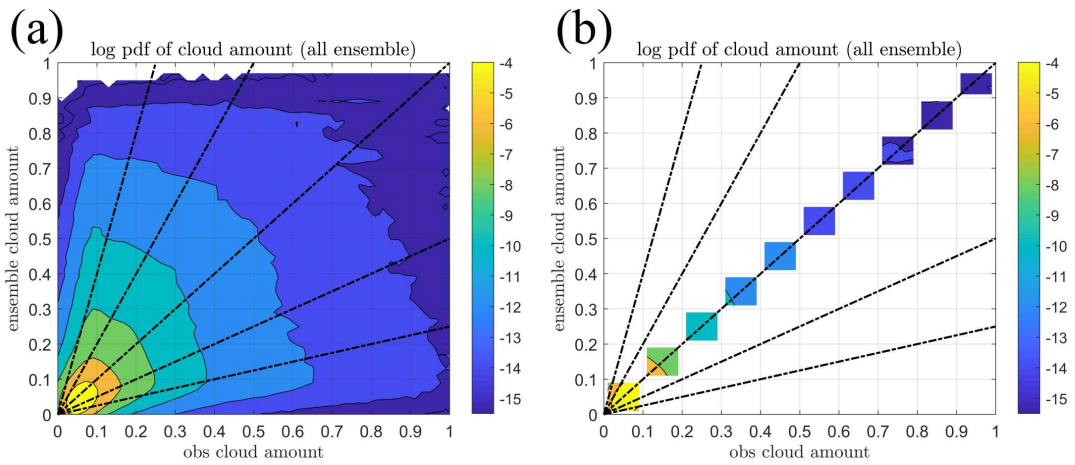

Figure 4.5: The logarithm of the joint distribution of the observed cloud amount and the ensemble cloud amount. The bin-size of the joint-histogram is 0.02. The black dashed lines are the 1-4, 1-2, 1-1, 2-1, and 4-1 line for reference. (a) for all ensemble members (b) for the ensemble members in the indistinguishable subset (see text).

4.4.2 A non-parametric state-dependent observation error model

We examine the cloud amount dependent observation error model constructed based on Equation (4.16) in this section. We divide the cloud amount into 10 categories, from 0 to 1 and with bin size 0.1. Figures 4.6-4.7 show the pdf f_{ε^o} for each cloud amount category, and all the pdfs used in DOEE in Equation (4.15). Note that for the mostly clear sky case (i.e., the cloud amount being [0.0,0.1]), the pdf is right-skewed and the tail is long at the positive values (Fig.4.6a). We attribute this right-skewness to the cloud amount asymmetry between the observation and the ensemble in this category (Note that this asymmetry only occurs in the mostly clear category. Specifically, if we compare the joint distribution of the observed cloud amount and the cloud amount in the indistinguishable subset (Fig.4.5b), we can see that except for the mostly clear category [0.0,0.1], the pdf is mostly symmetric in the other (cloudy) blocks). In this category, the joint histogram has larger mass in the region where the observed cloud amount is larger than the ensemble cloud

amount (Fig.4.5b). In addition, in this channel, the brightness temperature in general increases with the increasing cloud amount (Fig.4.8). Therefore, if the observation has larger cloud amount compared to the ensemble, the brightness temperature of the observation will be larger than the ensemble, leading to positive observation errors.

In general, the standard deviation of the observation error pdf increases with cloud amount when the cloud amount is between 0.0-0.5 (Figs.4.6a-e), and it remains constant or slightly decreases with cloud amount when the cloud amount is between 0.5 to 1.0 (Figs.4.6f-j). This result is qualitatively consistent with Geer and Bauer (2011), but quantitatively the standard deviations are smaller here (around 5 K here compared to 15–20 K there). The reason is that Geer and Bauer (2011) estimate the errors from the strong-constraint 4DVar perspective, where the observation error includes the errors in the forecast model during the assimilation window, while here our estimations do not include the errors in the forecast model. We will elaborate on the differences between these two errors in Section 4.5.

The mean of the observation error pdf, in generally, increases with cloud amount as well. Most of the pdfs are not Gaussian-like, and especially some are skewed, for example, [0.0, 0.1], [0.2, 0.3], and [0.4, 0.5]. This suggests that Gaussian assumption for the observation errors used in many current data assimilation schemes can be problematic when assimilating the satellite radiances.

We note that the cloud amount is just one choice for the predictor for the state-dependent observation error model for satellite radiances. However, we realize that the true observation error model can be more complicated. Therefore, we also examine a slightly more complicated state-dependent observation error model by adding the total column water vapor (tcwv) as a new predictor. In other words, we have two predictors for the observation error model: the cloud amount defined in Equation (4.20) and the tcwv. Specifically, we further divide the pixels in each category of cloud amount into two: the tcwv being larger or smaller than 40 kg m^{-2} in the high resolution control run (Since tcwv for each ensemble member and the observation are not available in the current dataset, we only use tcwv in the high-resolution control run. However, the variations of tcwv are believed to be small, so the tcwv in the control run should be representative enough).

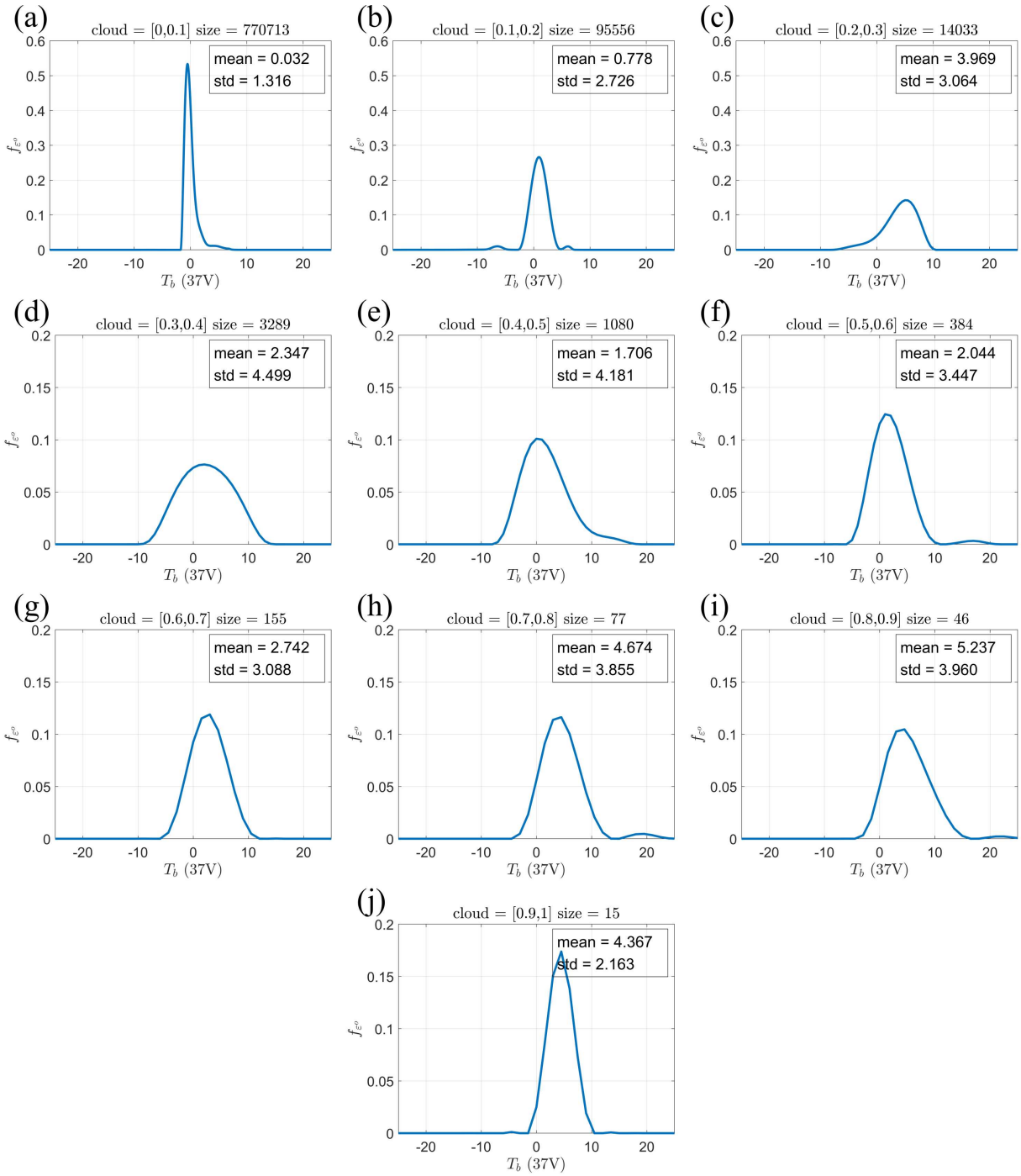

Figure 4.6: The cloud amount dependent observation error model for satellite radiance at 37 GHz vertical polarization, for cloud amount (a) [0.0,0.1], (b) [0.1,0.2], (c) [0.2,0.3], (d) [0.3,0.4], (e) [0.4,0.5], (f) [0.5,0.6], (g) [0.6,0.7], (h) [0.7,0.8], (i) [0.8,0.9], and (j) [0.9,1.0].

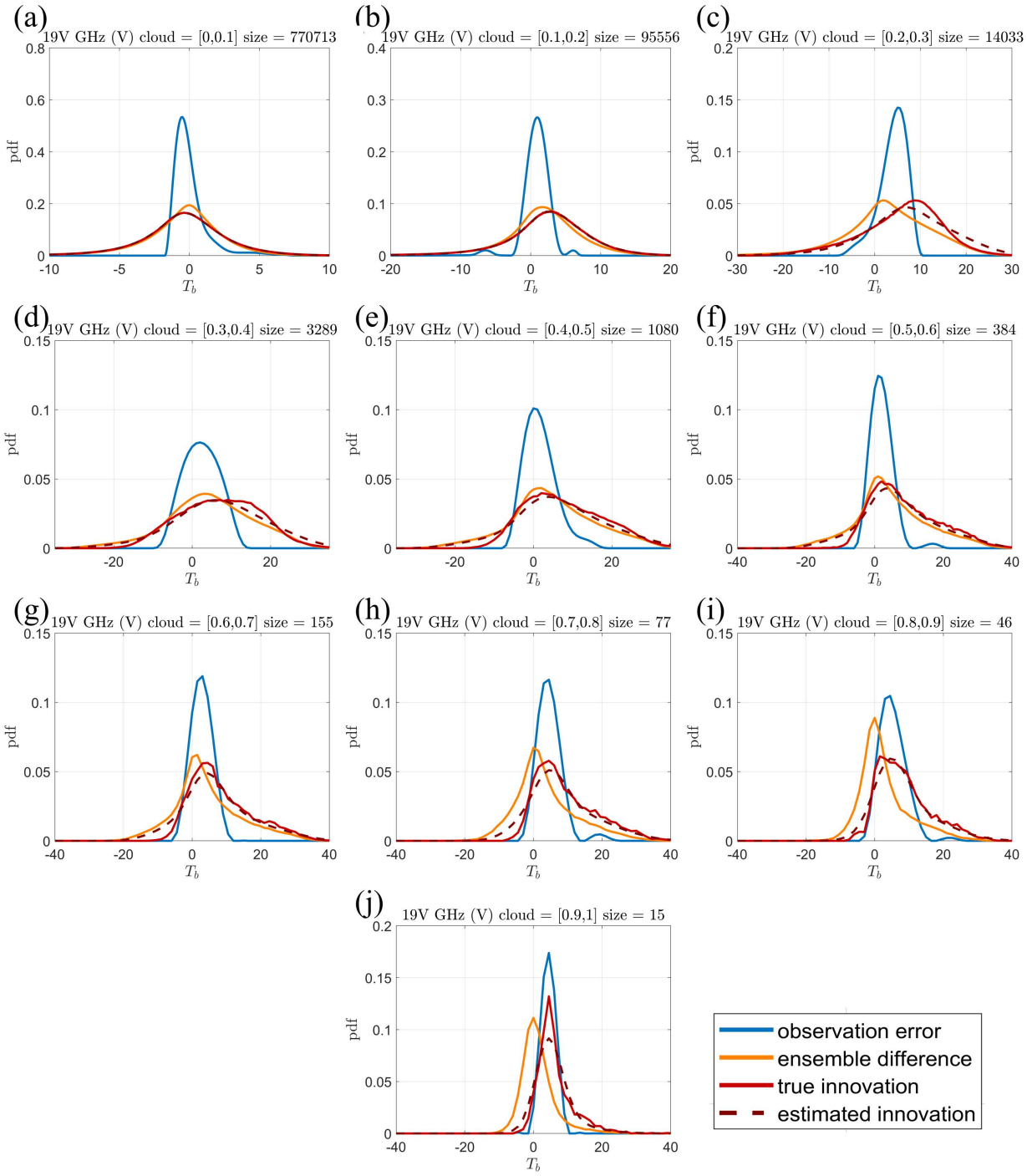

Figure 4.7: Similar to Figure 4.2, but for the cloud amount dependent observation error model for satellite radiance at 37 GHz vertical polarization. The blue line in each subplot is exactly the same as in Figure 4.6.

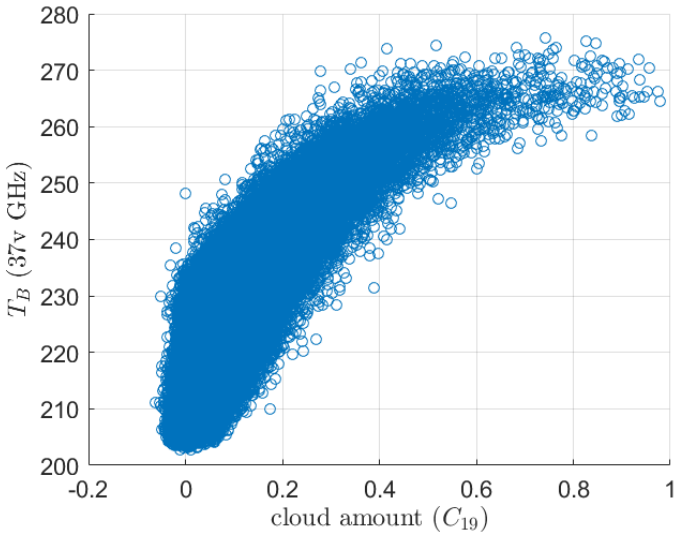

Figure 4.8: The relation between the observed cloud amount defined by the 19 GHz normalized polarization difference in Equation (4.20) and the observed radiance at 37 GHz vertical polarization brightness temperature.

The reason for choosing 40 kg m^{-2} as the criteria is based on the previous studies showing that 40-60 kg m^{-2} is the threshold for tropical convection (e.g., Bretherton et al., 2004; Schiro et al., 2016), which we believe could have very different observation errors properties compared to other cloud types.

Figure 4.9 shows the observation error pdfs as a function of cloud amount and tcwv. We first note that if the observation error pdf is only a function of cloud amount, then we can expect that the pdfs for tcwv<40 (blue curves in Fig.4.9) is similar to the pdfs for tcwv>40 (red curves in Fig.4.9) in each category, which is however not the case here. On the other hand, the observation error pdf is not only a function of tcwv either. If it is only a function of tcwv, then all the red curves in Figure 4.9 should be similar, and the same for all the blue curves. This suggests that the true state-dependent observation error model for radiances at this channel is indeed more complicated. Constructing a more accurate state-dependent observation error model is important as we can better utilize the information from the observations.

It is interesting to find that the positive tail of the observation errors in the mostly clear sky category (cloud amount [0.0,0.1]) is largely associated with pixels with smaller tcwv. Specifically,

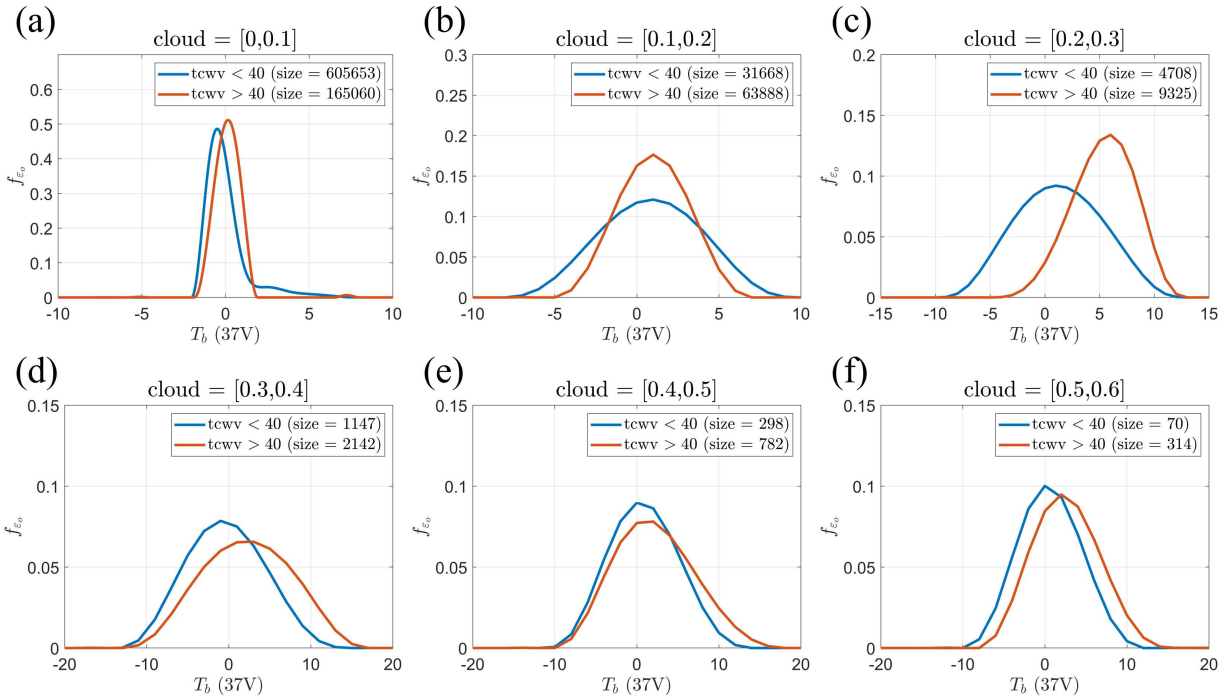

Figure 4.9: Similar to Figure 4.6, but the samples in each cloud amount category are further divided into two based on the total column water vapor (tcwv). The blue curve is the observation error pdf when tcwc < 40 (kg m^{-2}), and the red curve is when tcwv > 40 (kg m^{-2}). Note that only the pdfs for cloud amount smaller than 0.6 are shown, since there are less than 10 samples for tcwv < 40 in the cloud amount category when the cloud amount is larger than 0.6. The error model for cloud amount being (a) [0.0,0.1], (b) [0.1,0.2], (c) [0.2,0.3], (d) [0.3,0.4], (e) [0.4,0.5], (f) [0.5,0.6].

Figure 4.9a shows that when tcwv<40, there is a similar right-skewness in the observation error pdf (Fig.4.6a), while when tcwv>40, the observation error pdf is symmetric and close to Gaussian (Fig.4.9a). For cloud amount [0.1,0.2], the observation error pdf for tcwv<40 is wider than tcwv>40, and both of their mean are slightly larger than 0. This is consistent with Figure 4.6b.

The difference between the pdfs for tcwv<40 and tcwv>40 becomes clear when the cloud amount is larger than 0.2 (Figs.4.9c-f). It is interesting to see that, when tcwv<40, the pdfs for cloud amount larger than 0.2 (blue curves in Figs.4.9c-f) are very similar: they are symmetric, Gaussian-like, and centered close to 0, whereas when tcwv>40, the pdfs (red curves in Figs.4.9c-f) are very different but all of them have mean larger than 0. Note that the pixels with cloud and tcwv>40 are likely to be associated with convective clouds. These results at least suggest two things: first, the non-Gaussianity and skewness in the observation error for cloud amount larger than 0.2 (Figs.4.6c-f) can be mostly attributed to the pixels with larger tcwv. Second, there are biases in the pixels with tcwv>40 when cloud amount is larger than 0.2. As discussed in Section 4.3.2, we cannot distinguish where the biases originate by simply looking at the results: they might either come from the model biases, or from the errors in the observation operators. Note that we only examine the pdfs up to cloud amount being 0.6 in Figure 4.9 since the sample size for the sub-category tcwv<40 becomes too small when cloud amount is larger than 0.6.

4.5 Concluding remarks

In this study, we propose a new methodology, the Deconvolution-based Observation Error Estimation (DOEE), to estimate the full probability density function (pdf) of the observation error. DOEE relaxes two important assumptions used in previous methods: Gaussian assumptions for the errors and the linear approximation for the observation operator. The new method can theoretically retrieve any form of the observation error pdf based on the statistics of the observations and an ensemble of model equivalences. We demonstrate a series of idealized experiments in Section 4.3, and find that DOEE can successfully retrieve the observation error pdf when it is either a Gaussian or a non-Gaussian (bimodal, bounded or skewed) distribution.

The key equations we use in the DOEE are Equations (4.10) and (4.15). To have an estimate of the pdf of the simulated truth in the observation space $H\left(X^{truth}\right)$, we propose to randomly pick an ensemble member as a sample for $H\left(X^{truth}\right)$. This is based on the assumption that the ensemble is indistinguishable from the truth (the i.i.d assumption). Note that the i.i.d assumption is also implicitly used in ensemble methods (see the Appendix C.1 for details), so we are not introducing an extra assumption here. In addition, we can construct a state-dependent observation error model based on Equation (4.16). In this case, instead, we propose to select the ensemble member from the "indistinguishable subset", which is a subset of the ensemble with a certain value or a certain range of the predictor $C\left(X\right) \in \mathcal{C}$, to sample $H\left(X^{truth}{}_{|C\left(X^{truth}\right)\in\mathcal{C}}\right)$.

We applied the DOEE method to construct state-dependent observation error models for the satellite radiances at 37 GHz vertical polarization, based on Equation (4.16). We first construct a cloud amount dependent observation error model. The results shown in Figure 4.6 suggest that: (1) The standard deviation of the pdf generally increases with cloud amount. (2) Many of the pdfs are skewed and non-Gaussian. Furthermore, we examine a slightly more complicated state-dependent observation error model by adding the total column water vapor (tcwv) as a second predictor, and the results are shown in Figure 4.9. What we can learn from this slightly more complicated model are: (1) The cloud amount alone may not be the optimal predictor for the observation error model, since the pdfs for tcwv<40 and tcwv>40 at the same cloud amount category can be very different. (2) For mostly clear sky category (cloud amount [0.0,0.1]), the non-Gaussianity in the observation error pdf is associated with the pixels with tcwv<40, whereas for the cloudy categories (cloud amount 0.2-0.6), the non-Gaussianities are associated with the pixels with tcwv>40. (3) For cloudy categories (cloud amount 0.2-0.6), the pdfs for tcwv<40 are similar, which might suggest the observation errors in these pixels come from the same distribution. In contrast, the pdfs for tcwv>40 differ for different cloud amount category, while all of them have positive mean. This suggests there are some biases in the simulated radiances from these cloudy pixels with tcwv>40. Exploring other predictors for the observation error models, like cloud type, would be an interesting topic

for the future work. Note that with a better set of predictors for the state-dependent observation error model, we can better utilize the information from the observations.

We should emphasize that, in Section 4.4, we concentrate on the observation operator from a filter perspective. If, instead, a strong-constraint smoother is used, such as the strong-constraint 4DVar, the representation errors will also include errors in the model equations. To clarify, we denote the radiative transfer model (RTM) as H and the model forecast from the beginning of the window to the observation time as M. For the observation error of RTM alone, we are looking at the errors in H, while for the observation error from the strong-constraint 4D-Var perspective, we are looking at the errors in the composite function $H \circ M$. There is no fundamental difference in estimating the observation error for either H or $H \circ M$ using DOEE. However, when constructing a state-dependent model in Equation (4.16), the assumption that $C\left(X^{truth}\right) \approx C_{obs}$ may not always hold. For example, when it applies to constructing a similar cloud amount (but at the observation time) dependent error model for $H \circ M$, the assumption that $C\left(M\left(X^{truth}\right)\right) \approx C_{obs}$ becomes less appropriate because we do recognize that the model M typically has large errors in predicting clouds. Therefore, it becomes difficult to construct a state-dependent observation error model for $H \circ M$ when the predictor is the cloud amount at the observation time. Instead, we may choose the tcwv at the observation time as a predictor and assume that tcwv $\left(M\left(X^{truth}\right)\right) \approx$ tcwv$_{obs}$ since the model will typically predict tcwv more accurately than the cloud amount. Exploring a proper predictor for the state dependent observation error models from the strong-constraint 4D-Var perspective is left for future work.

It is also important to note the assumption that the background error and the observation error are independent can be problematic for some cases, especially for the observation error model of the all-sky radiances from the strong-constraint 4D-Var perspective. While it is possible in principle to include dependent background and observation errors, the innovation pdf will not be a simple convolution anymore, and more work is needed to extend DOEE to this case.

Finally, we briefly comment on ways to include the non-parametric observation errors in some current data assimilation methods. A data assimilation method that does not require a parametric

form of the observation error, for example, particle filters and particle flow filters (e.g., Daum and Huang, 2011; Ades and Van Leeuwen, 2015; Poterjoy, 2016; Pulido and van Leeuwen, 2019; van Leeuwen et al., 2019; Hu and van Leeuwen, 2021, etc) or the quantile-conserving filter (Anderson, 2022), would be natural choices to assimilate observations with non-parametric observation errors. For methods that require a parametric form for the observation error, we may, for example, locally approximate the non-parametric pdf by the required parametric pdf. If the observation error is state-dependent and we are using a variational method, we may need to re-approximate the non-parametric pdf by a different Gaussian in each outer loop. In ensemble Kalman filters, the observation error variance should be the expected value of $\left(\varepsilon^o\right)^2$ over the prior distribution (Bishop, 2019). In this case, we have to estimate $\left(\varepsilon^o\right)^2$ for the pdf in each category, and average these values based on the prior distribution of the predictors.

In all, DOEE is an important methodological step forward in diagnosing the non-Gaussianity of the observation error. In addition to the non-Gaussianity, the state-dependent error, spatial and temporal correlations of the error, and the separability of the background and the observation error are also important aspects of the observation error. Although it's not clear that whether the non-Gaussianity is the main challenge that needs priority to be addressed in a numerical weather prediction (NWP) system, the diagnosed non-Gaussianity in real observations is clear. We are currently implementing DOEE and testing the inclusion of the non-Gaussian observation error in a full-scale NWP system. The impact of the non-Gaussianity on the DA and NWP will be explored in future studies.

Chapter 5

Introducing non-Gaussian observation errors into variational methods

5.1 Background and motivation

Although the existence of non-Gaussianities in the observations errors have been recognized for a while (e.g., Pires et al., 2010), there are two main obstacles that prevent us from incorporating these non-Gaussianties into DA. The first one is that the actual shape or the parametric form of the observation error is typically unknown, mainly due to the presence of the representation error. The Deconvolution-based Observation Error Estimation (DOEE) that we propose in Chapter 4 can estimate a non-parametric form of the observation error, and therefore DOEE is a solution to this problem. However, here comes the second obstacle: how do we actually use the information from the non-Gaussian errors? The issue is that most of the modern DA schemes used in the operational centers, e.g., 4D-Var, still rely on Gaussian assumptions for the observation error. Finding a general and efficient way to incorporate the non-Gaussian errors into variational methods is the main focus of this work.

Despite the fact that the actual shape of the non-Gaussian observation error is unknown before DOEE, there have already been a few developments that try to tackle the challenges that arise from the non-Gaussianities based on some assumed parametric distributions. We will first briefly review these developments and then point out the unique contribution from this work. These developments are mainly used to address the issues from "gross error". The gross error is typically defined by the observation that departs from the expected value by more than a few standard deviations, which is considered as a rare event for a Gaussian error. The gross error could happen due to the presence of the representation error as mentioned, as discussed in the previous chapter. Specifically, it has been found that even for many in-situ conventional observations, their O-B

innovations distribution exhibit more frequent gross error compared to a Gaussian, and therefore suggesting a non-Gaussian observation error distribution with heavier tails (e.g., Tavolato and Isaksen, 2015). Assimilating the observation with gross error can sometimes lead to poor analysis due to over-fitting if using Gaussian observation error (Lorenc, 1984). One simple way to overcome this issue is to conduct quality control before DA. For example, one can discard the observation that is farther away than 3 times of the observation error standard deviation than the background, so that the heavy tail can be reduced. Instead of simply discarding the outlier observations, the European Centre for Medium-Range Weather Forecasts (ECMWF) has implemented a variational quality control (varQC) (Anderson and Järvinen, 1999), which assumes that the observation error distribution is "Gaussian plus flat". Specifically, the observation error is the sum of a Gaussian and a uniform distribution. Adding a uniform distribution is based on the notion that there is no information about the presence of the gross error, and this has the effect of making the Gaussian tail heavier. Mathematically, it can be shown that using the "Gaussian plus flat" essentially assigns weight to each observation. The weight is a function of the difference between the observation and the background: when they are close the weight is 1 (so the assumed error is purely Gaussian), and when they become farther away the weight smoothly decreases from 1 to 0 (0 means totally discard the extreme outlier). Therefore, the "Gaussian plus flat" can smoothly down-weight and eventually discard the outliers, instead of discarding all the observation whose value is outside a certain threshold.

However, Tavolato and Isaksen (2015) points out that the transition of the weight from 1 to 0 is still too sharp, and they also show that the "Gaussian plus flat" distribution does not well represent the tail of the O-B innovation distribution for some conventional observations. Instead, they proposed to use the Huber norm (Huber, 1972), which is a distribution with a Gaussian shape in the central connected with two exponential distributions in the tails. The Huber norm is a more flexible distribution, allowing a smoother transition of the weight for the observation, and can be used to fit the O-B innovation better for the conventional observations. Tavolato and Isaksen (2015) also shows that using a Huber norm can better assimilate observations in extreme events, e.g., the

dropsonde wind and temperature observations within tropical cyclones, leading to a better analysis compared to using "Gaussian plus flat".

As mentioned earlier, many of the above developments are based on an assumed form for the observation error. How accurate these assumed distributions can represent the actual observation error is less clear. In addition, we note that Huber norm is still restrictive in representing the shape of an arbitrary distribution. Now that we have access to the shape of the non-parametric observation error, we aim to better utilize the information in this distribution. Therefore, the unique contribution from this work is that we propose a general framework that can incorporate the non-Gaussianities, whether the distribution is parametric or non-parametric, into the variational methods (specifically, the incremental form). Moreover, we do not require revising the Gaussian cost-function in the variational methods in this framework, thereby ensuring its efficiency as a method without adding extra computational cost.

The remaining of this paper is organized as follows. Section 5.2 describes the new proposed framework, the evolving-Gaussian method, that incorporates non-Gaussianities into variational methods. Section 5.3 demonstrates the evolving-Gaussian method and shows it can improve the assimilation over purely Gaussian methods in an idealized experiment. Then, we implement the evolving-Gaussian method into a full-scale operational system, the Integrated Forecasting System (IFS) in ECMWF in Section 5.4. Lastly, Section 5.5 summarizes the findings in this work and presents future research directions.

5.2 Methodology

In this section, we first review the variational method and its incremental form in Section 5.2.1. We will look at how the variational methods deal with the observation error in Section 5.2.2. Then, we will present the new method, the evolving-Gaussian method, to introduce the non-Gaussianties into the variational method in Section 5.2.3.

5.2.1 Variational data-assimilation methods

We start this section with a brief overview of the theoretical background of variational DA methods. Specifically, we will focus on the "incremental form" of the variational method (e.g, Courtier et al., 1994; Bonavita et al., 2018). The basic idea of the variational DA methods is to find the mode of the posterior probability density function (pdf). This is done by first formulating a cost-function, which is the negative of the logarithm of the posterior pdf, and then find the state that minimizes the cost-function in Equation (5.1):

$$\begin{aligned} J(\mathbf{x}) &= -\log p(\mathbf{x}|\mathbf{y}_o) \\ \mathbf{x}_a &= \underset{\mathbf{x}}{\operatorname{argmin}}\, J(\mathbf{x}) \end{aligned} \tag{5.1}$$

where $p(\mathbf{x}|\mathbf{y}_o)$ is the posterior pdf, and $\mathbf{x}_a$ is the analysis for 4D-Var.

To find the minimum of the cost-function J in Equation (5.1), we can, e.g., use gradient descent to find the solution. However, gradient descent can be inefficient, especially when the dimension of $\mathbf{x}$ is high, e.g., $O(10^{10})$ in a full-scale operational weather forecast system. We can use Newton method to accelerate the convergence, while Newton method requires the inverse of the Hessian of J, which is not possible for such a high-dimensional system. Instead, an alternative way, that avoids the calculation of the inverse of Hessian, is to explore the Gauss-Newton method, which is the essence of the incremental 4D-Var. In the incremental 4D-Var, finding the solution in (5.1) is replaced by sequentially finding the minimum of quadratic cost-functions that are easier to minimize, which is obtained by linearizing the original cost-function around a specific state.

In the incremental 4D-Var, we start with the background state, or the mean of the prior pdf, i.e., $\mathbf{x}^0 = \mathbf{x}_b$. First, we linearize the cost-function J at around $\mathbf{x}^0$ to form a quadratic cost-function Q^0. Specifically, let $J(\mathbf{x}) = \frac{1}{2}\mathbf{f}(\mathbf{x})^T\mathbf{f}(\mathbf{x}) = \frac{1}{2}\|\mathbf{f}(\mathbf{x})\|^2$ [3] and

$$Q^0(\Delta\mathbf{x}) \equiv \frac{1}{2}\|\mathbf{f}(\mathbf{x}^0) + \nabla\mathbf{f}(\mathbf{x}^0)\Delta\mathbf{x}\|^2 \approx \frac{1}{2}\|\mathbf{f}(\mathbf{x}^0 + \Delta\mathbf{x})\|^2 \tag{5.2}$$

[3]Note that $\mathbf{f}$ is not the particle flow discussed in Chapter 2-3. See Appendix D.1 for how $\mathbf{f}$ is found for details.

where $\nabla \mathbf{f}(\mathbf{x}^0)$ is the Jacobian matrix of $\mathbf{f}$ evaluated at $\mathbf{x}^0$. Then, we find $\Delta \mathbf{x}$ that minimizes Q^0, and update the state as $\mathbf{x}^1 = \mathbf{x}^0 + \Delta \mathbf{x}$. Note that the advantage of the quadratic cost-function in Equation (5.2) is that there are efficient algorithms to find $\Delta \mathbf{x}$. After obtaining an updated state $\mathbf{x}^1$, we linearize the cost-function J in Equation (5.1) again but at around the updated state $\mathbf{x}^1$. We get an updated quadratic function Q_1 as follows:

$$Q^1(\Delta \mathbf{x}) \equiv \frac{1}{2} \|\mathbf{f}(\mathbf{x}^1) + \nabla \mathbf{f}(\mathbf{x}^1) \Delta \mathbf{x}\|^2 \tag{5.3}$$

and similarly, we use an efficient algorithm to find $\Delta \mathbf{x}$ that minimizes Q^1 and update the model state as $\mathbf{x}^2 = \mathbf{x}^1 + \Delta \mathbf{x}$. We repeat the above procedure to generate a sequence of quadratic cost-functions $\{Q^0, Q^1, ...Q^n\}$ and a sequence of model states $\{\mathbf{x}^0, \mathbf{x}^1, ...\mathbf{x}^n\}$. The solution converges when $\mathbf{x}^n - \mathbf{x}^{n-1}$ is small. The final state $\mathbf{x}^n$ is then the analysis of the 4D-Var, i.e., $\mathbf{x}_a = \mathbf{x}^n$.

Note that the iterations of formulating the quadratic cost-functions Q^i are part of the "outer loop". The iterations (from any efficient algorithm) used to find $\Delta \mathbf{x}$ for each quadratic Q^i form the "inner loop". For example, the 4D-Var used at ECMWF uses 3 outer loops, and around 30-40 inner loops within each outer loop.

5.2.2 The cost-function for Gaussian observation errors

The observation error is assumed to be Gaussian in 4D-Var, so that the cost-function and its gradient can be easily derived. Specifically, we can decompose the cost-function in Equation (5.1) into two components based on Bayes theorem:

$$J(\mathbf{x}) = J_b(\mathbf{x}) + J_o(\mathbf{x}) \tag{5.4}$$

where $J_b(\mathbf{x}) = -\log p(\mathbf{x})$ is related to the prior pdf and $J_o(\mathbf{x}) = -\log p(\mathbf{y}_o|\mathbf{x})$ is related to the observation error and the likelihood function. With the assumption of Gaussian observation error, J_o can be written as

$$J_o(\mathbf{x}) \propto \frac{1}{2}(\mathbf{y}_o - H(\mathbf{x}))^T \mathbf{R}^{-1}(\mathbf{y}_o - H(\mathbf{x})) \tag{5.5}$$

where $H(\mathbf{x})$ is the simulated observation vector or the model equivalence of the observations, and $\mathbf{R}$ is the observation error covariance. Note that it is typically assumed that the observation errors are independent, i.e., $\mathbf{R}$ is a diagonal matrix. In that case, Equation (5.5) can be re-written as

$$J_o(\mathbf{x}) \propto \frac{1}{2} \sum_j \left(\frac{y_{o,j} - H_j(\mathbf{x})}{\sigma_{o,j}} \right)^2 \tag{5.6}$$

where $y_{o,j}$ is the j-th scalar observation and H_j is the corresponding observation operator, and $\sigma_{o,j}$ is the observation error standard deviation for the j-th observation. In this case, it can be shown that assimilating each observation sequentially (and the order does not matter) and assimilating all the observations at once will give the same result. Therefore, for convenience, in the following discussion we only focus on the assimilation of a scalar observation, i.e., J_o is further simplified as

$$J_o(\mathbf{x}) \propto \frac{1}{2} \left(\frac{y_o - H(\mathbf{x})}{\sigma_o} \right)^2 \tag{5.7}$$

We drop the subscript j since there is only one observation. It can be inferred from Equation (5.2) that linearizing the cost-function requires the gradient of J, and hence the gradient of J_o, which is

$$\nabla J_o(\mathbf{x}) = -\mathbf{H}^T \left(\frac{y_o - H(\mathbf{x})}{\sigma_o^2} \right) \tag{5.8}$$

where $\mathbf{H}^T = \nabla H(\mathbf{x})$ is the linearized observation operator.

One of the strength of the incremental 4D-Var in assimilating nonlinear observations (e.g., Bonavita et al., 2018) lies in Equation (5.8). The reason is as follows. Observe that $\mathbf{H}^T$ depends on $\mathbf{x}$ when H is nonlinear. Although $\mathbf{H}^T$ is a constant (matrix) in the inner loop, it is updated when we linearize the cost-function to obtain a new quadratic cost-function in the next outer loop. In other words, in the very first outer-loop, the linearized quadratic cost-function Q^0 may only locally approximate the true cost-function J at around the first guess state $\mathbf{x}^0$, whereas Q^0 and J

may be very different elsewhere (because $\mathbf{H}^T$ is not uniform). As $\mathbf{x}^n$ gradually becomes closer to the minimum and the re-linearization of the cost-function in later outer loops, we can expect an improved approximation of Q^n to J at around the true minimum.

5.2.3 The evolving-Gaussian method for non-Gaussian observation errors

For non-Gaussian observation errors, one can try to derive the cost-function and its gradient as in Section 5.2.2, if we know the parametric form. For example, the varQC and Huber norm used in ECMWF as mentioned in the Introduction are based on this strategy. In addition, Fletcher and Zupanski (2006a) derived the variational method with a log-normal distribution, which has implications for positive definite variables, e.g., humidity.

Although it is possible to derive the cost-function and its gradient for parametric observation errors, we do not always know the parametric form of the observation error, especially for observations with large representation error. One solution is to use the Deconvolution-based Observation Error Estimation (DOEE) in Chapter 4 to identify the parametric form of the observation error, and then derive the cost-function and its gradient. However, one can still envisage a case where the estimated non-parametric pdf is far from any parametric pdfs.

The solution we are looking for here is a more general one. Note that we start from a non-parametric pdf of the observation error in the first place. The idea is to take the advantage of the outer loop, where we not only re-linearize $\mathbf{H}^T$ in J_o but we also re-approximate the non-Gaussian observation error pdf with a different Gaussian pdf in each outer loop. Mathematically, we can see this in the following. First recall the definition of J_o for a single observation (without Gaussian assumptions) is:

$$J_o(\mathbf{x}) = -\log p(y_o|\mathbf{x}) = -\log p(y_o|H(\mathbf{x})) \tag{5.9}$$

We note that Equation (5.7) is a special case where we explicitly assume Gaussian errors for Equation (5.9). For simplicity, if assuming linear H for now, we note that Equation (5.7) is already quadratic. This is because of the shape of the Gaussian pdf. In other words, if the observation

error in Equation (5.9) is non-Gaussian, the likelihood cost-function J_o is still non-quadratic even for linear H. Nevertheless, we can adopt a similar idea as we did in making the cost-function quadratic: we use a Gaussian error as in Equation (5.7), but we update the Gaussian (i.e., with different mean and standard deviation) in each outer loop to account for the non-Gaussianity. Since we "evolve" the Gaussian for the observation error in each outer loop, we call this method the "evolving-Gaussian" method. This can be easily generalize to non-linear H, where we both update $\mathbf{H}^T$ and the Gaussian in each outer loop to account for the non-linearity and the non-Gaussianity.

The question now reduces to how we choose the mean and standard deviation of the Gaussian to locally approximate J_o in each outer loop. The strategy that we adopt is as follows:

1. Choose the mean of the Gaussian equal to the mode of J_o. Note that if the mode of the observation error is found at 0, then the mode of J_o is at the observed value y_o. In general, if the mode of the observation error is $m \neq 0$, then we set the mean of the Gaussian equal to $y_o - m$.

2. Calculate the standard deviation of the approximated Gaussian σ_o based on the local gradient of J_o with respect to $H(\mathbf{x})$. Specifically, we ensure that the local gradient of the Gaussian cost-function is the same as the non-Gaussian cost-function, i.e.,

$$\nabla_{H(\mathbf{x})} J_o = \nabla_{H(\mathbf{x})} \left\{ -\log p(y_o|H(\mathbf{x})) \right\} \equiv \nabla_{H(\mathbf{x})} \left\{ \frac{1}{2} \left(\frac{(y_o - m) - H(\mathbf{x})}{\sigma_o} \right)^2 \right\} \tag{5.10}$$

   We can rearrange the above equation to obtain:

$$\sigma_o = \sqrt{\frac{(y_o - m) - H(\mathbf{x})}{\nabla_{H(\mathbf{x})} \log p(y|H(\mathbf{x}))}} \tag{5.11}$$

3. In each outer loop, update σ_o based on the local gradient of $\nabla_{H(\mathbf{x})} \log p(y_o|H(\mathbf{x}))$ at the updated state.

It is worth noting that if it is *affordable* to (but it's not, see Section 5.2.1) use gradient descent to find the solution in Equation (5.1), then we don't need the evolving Gaussian method because

$$\begin{aligned}\nabla_{\mathbf{x}} J(\mathbf{x}) &= \nabla_{\mathbf{x}} J_b(\mathbf{x}) + \nabla_{\mathbf{x}} J_o(\mathbf{x}) \\ &= \nabla_{\mathbf{x}} J_b(\mathbf{x}) - \nabla_{\mathbf{x}} H(\mathbf{x})[\nabla_{H(\mathbf{x})} \log p(y_o|H(\mathbf{x}))]\end{aligned} \tag{5.12}$$

where we can directly derive the value of $\nabla_{H(\mathbf{x})} \log p(y_o|H(\mathbf{x}))$ based on the results from DOEE. We might also just plug in the number of $\nabla_{H(\mathbf{x})} \log p(y_o|H(\mathbf{x}))$ and update its value in each inner loop, but this makes the linearized cost-function non-quadratic again. The restriction here is that we want to make the cost-function quadratic to ensure the numerical stability of the whole 4D-Var (e.g., the inner loop might require much more iterations to converge if the cost-function is non-quadratic, which increases the computational cost). It would be interesting to examine the impact of introducing some non-Gaussianities by directly using $\nabla_{H(\mathbf{x})} \log p(y_o|H(\mathbf{x}))$ in the inner loop, and compare that result with the one from the evolving Gaussian method. We leave that for future work.

5.3 A demonstration of the evolving-Gaussian method

In this section we first demonstrate how to use the evolving-Gaussian method (Section 5.2.3) if the observation error or the likelihood function is found to be non-Gaussian. Then, we demonstrate in an idealized experiment that using the evolving-Gaussian method will improve the analysis against other methods assuming purely Gaussian observation errors in the incremental 4D-Var.

5.3.1 An example of a Gamma likelihood

We demonstrate how to use the evolving-Gaussian method to approximate a Gamma likelihood $Gamma(2, 2)$ in this section. The mean of the approximated Gaussian is set to the mode of $Gamma(2, 2)$, which is 2 (i.e., $y_o - m = 2$). Note that we only specify the likelihood function here and there are infinitely many possible combinations for the observation and the observation error pdf for this to happen. For example, it can be the case that the actual observed value is 2

($y_o = 2$) and the mode of the observation error is 0 ($m = 0$). It can also be the case that the actual observed value is 0 ($y_o = 0$) and the mode of the observation error is -2 ($m = -2$). Which one is the actual case does not matter, because they all result in the same likelihood function. We then estimate the standard deviation of the approximated Gaussian σ_o as a function of $H(\mathbf{x})$ using Equation 5.11. Figure 5.1 shows that the estimated standard deviation σ_o increases monotonically as $H(\mathbf{x})$ increases. This feature reflects the long tail of $Gamma(2, 2)$ right of the mode, where the gradient of the pdf decreases slower than a Gaussian and therefore corresponds to an increasing σ_o.

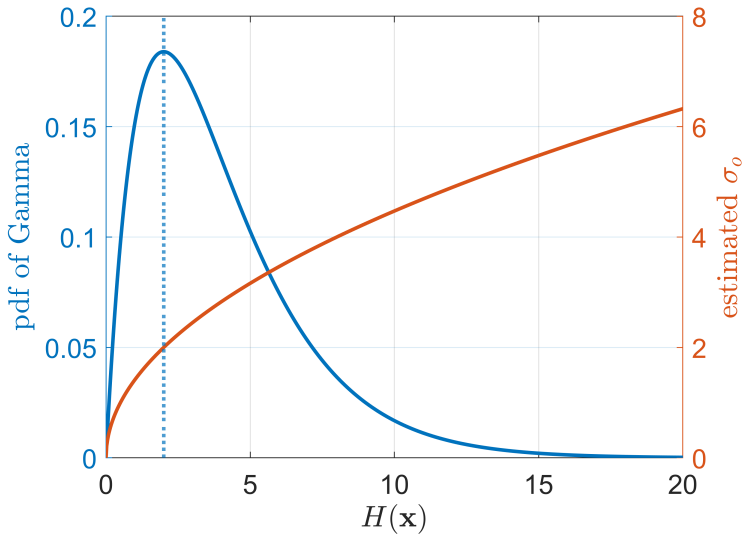

Figure 5.1: The $Gamma(2, 2)$ likelihood (blue) and the estimated standard deviation (red) as a function of $H(\mathbf{x})$. The vertical blue dashed line shows the mode of $Gamma(2, 2)$, which is set to the mean of the approximated Gaussian.

We can also look at the cost-function of the $Gamma(2, 2)$ likelihood and the estimated Gaussian at different values of $H(\mathbf{x})$ in Figure 5.2. Recall in Equation (5.10) that the standard deviation of the Gaussian is chosen such that the gradient of the cost-function for the non-Gaussian likelihood is the same the gradient of the cost-function of the Gaussian pdf. Therefore, the local gradient of the black line at the black dot should be the same as the gradient of the blue line at the blue dot. We can see that when the local gradient is larger at $H(\mathbf{x}) = 0.5$, we use a narrower Gaussian

(Figure 5.2a), whereas when the local gradient is smaller at $H(\mathbf{x}) = 5.0$, a wider Gaussian is used (Figure 5.2b). We will show in Section 5.3.3 that updating the Gaussian in each outer loop can improve the analysis of the incremental 4D-Var.

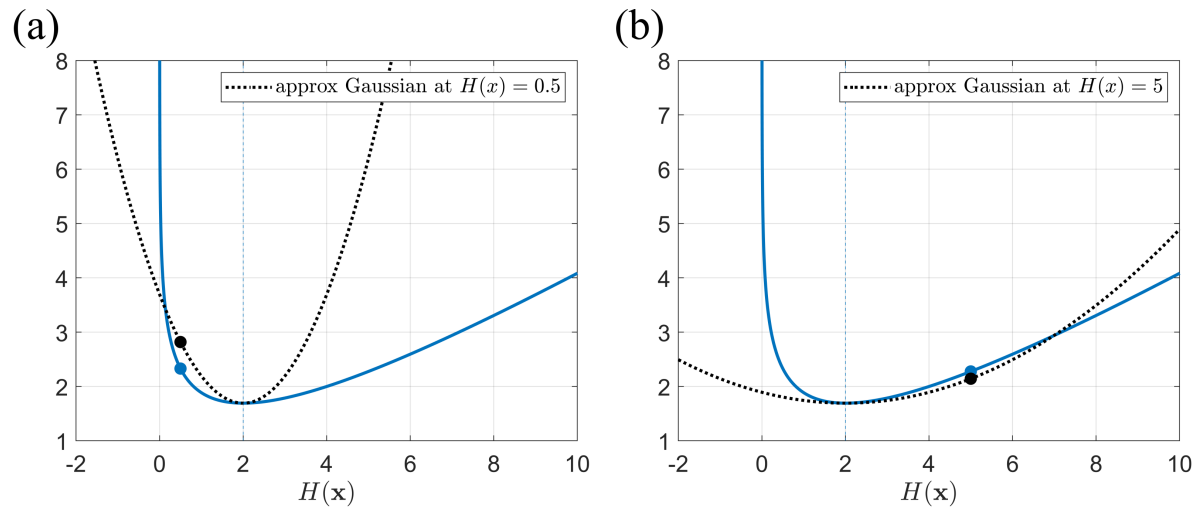

Figure 5.2: The cost-function of $Gamma(2, 2)$ likelihood (blue) and the approximated Gaussian (black dashed) at (a) $H(\mathbf{x}) = 0.5$ (b) $H(\mathbf{x}) = 5.0$.

5.3.2 A single-cycle DA experiment setup

We setup a simple DA experiment to see how the evolving-Gaussian method can improve the analysis of the incremental 4D-Var. We assume the model state is 1-D, and the observation operator is linear, so $H(x) = x$. Note that the results found here can be generalized to high dimensional state with linear H. With nonlinear H, the results should still apply if we have accurate $\mathbf{H}^T$ for each outer loop and the increment of state in each outer loop is small such that it falls within the linear regime of $\mathbf{H}^T$.

We conduct a series of a single cycle DA experiments. Each experiment starts with a different Gaussian prior, which all of them have the same variance ($\sigma_b^2 = 8$) but centered at different mean ranging from 0 to 15 (the gray curves in Figure 5.4a). All the experiments assimilate the same observation $y_o = 2$, with non-Gaussian observation error with mode at 0 ($m = 0$) such that the likelihood function is $Gamma(2, 2)$. We emphasize the difference between the observation error

pdf and the likelihood (as a function of $H(\mathbf{x})$) here. The observation error pdf is the distribution of $y - H(\mathbf{x})$, whereas the likelihood describes the distribution of $H(\mathbf{x})$ when a specific observation $y = y_o$ is seen, so it has the same shape as the distribution of $-(y - H(\mathbf{x}))$. This difference is not easily seen for symmetric observation error, e.g., Gaussian, but is obvious for non-symmetric distributions. The observation error and the likelihood as a function of $H(\mathbf{x})$ when $y_o = 2$ are shown in Figure 5.3.

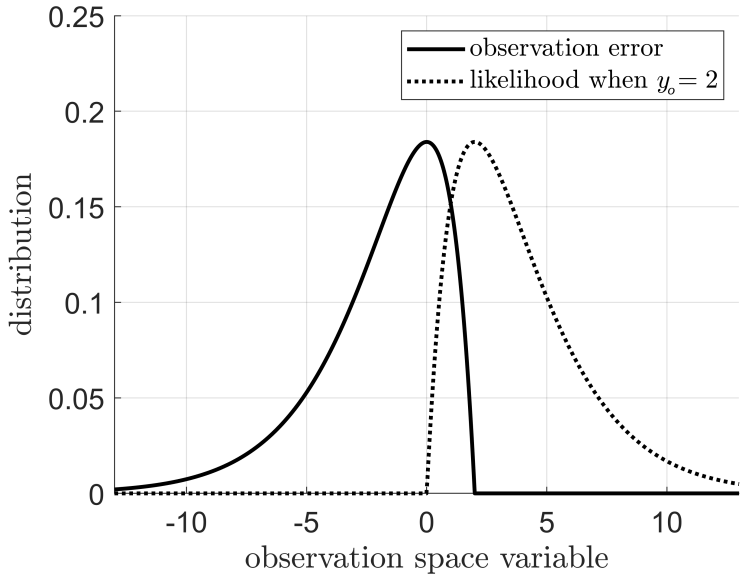

Figure 5.3: A demonstration of the difference between the observation error (solid) and the likelihood as a function of $H(\mathbf{x})$ when $y_o = 2$.

We will compare three different methods of treating the observation error and the likelihood:

1. Fixed Gaussian, with the mean and the standard deviation of the assumed Gaussian set to the mean and the standard deviation of $Gamma(2, 2)$. This is called "Gaussian-mean" experiment (the red curve in Figure 5.4b).

2. Fixed Gaussian, with the mean and the standard deviation of the assumed Gaussian set to the mode and the standard deviation of $Gamma(2, 2)$. This is called "Gaussian-mode" experiment (the blue curve in Figure 5.4b).

3. Evolving Gaussian. This uses the new method described in Section 5.2.3, where the likelihood evolves in each outer loop (which depends on the prior value, so not shown in Figure 5.4b). This is called "evolving-Gaussian" experiment.

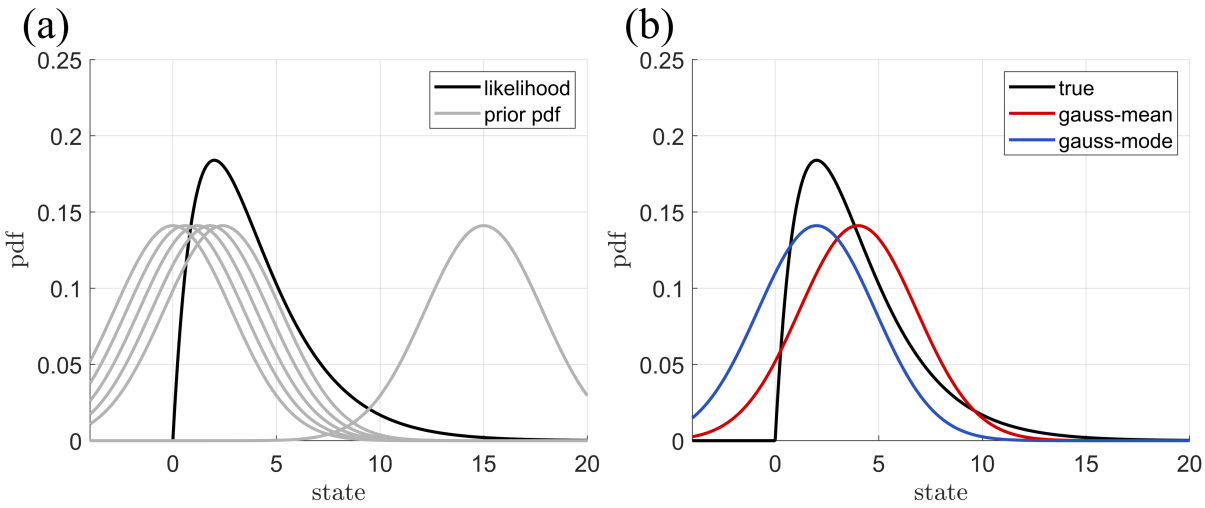

Figure 5.4: A demonstration of the series of single cycle DA experiment. The black curve is the true likelihood function. (a) The gray curves are the prior pdfs for each of the DA experiment (b) The blue and red curve is the assumed likelihood for "Gaussian mode" and "Gaussian mean" experiment respectively.

We note that the "Gaussian-mean" and "Gaussian-mode" experiments are designed to mimic a case in which we don't know the shape of the pdf (when DOEE is not available). In this case, assume we can only estimate the second-moment of the observation error pdf by using existing methods, e.g., background subtraction method (Andersson, 2003), Desroziers method (Desroziers et al., 2005), three-corned hat method (e.g., Anthes and Rieckh, 2018; Rieckh and Anthes, 2018), etc. Choosing the mean of observation error equal to the mean of $Gamma(2, 2)$ is to mimic the situation where we use a very simple bias correction scheme correct the observation such that the expectation of the observation error is equal to zero. (Since the mean of the observation error (Figure 5.3) is -2, the *equivalence* bias correction is to replace y_o by y_o+2 before the assimilation). Note that the mean and the mode are different for a skewed pdf. When we can only use Gaussian observation error, it is unclear whether using the simple bias correction scheme to enforce the mean

of the observation error to be zero is beneficial or not. Therefore, we also conduct the "Gaussian-mode" experiment to mimic a case where we do not bias correct the observation.

5.3.3 The result for the single-cycle DA experiment

The results for the series of single-cycle DA experiments is shown in Figure 5.5. Note that since the simple DA experiment is 1-D, so the posterior pdf can be found based on Bayes theorem and its mode (the gray dashed curve in Figure 5.5) can be easily identified. We can see that the solutions from the evolving-Gaussian experiments are almost identical to the analytical solutions (orange curve). On the other hand, the solutions from the Gaussian-mean (red curve) and Gaussian mode (blue curve) experiments are not always as good.

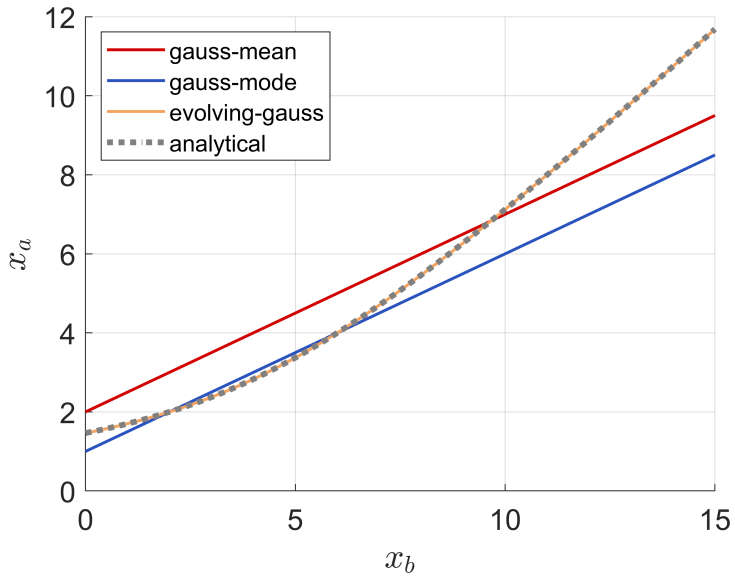

Figure 5.5: The analysis x_a of the incremental 4D-Var as a function of different background x_b. The gray dashed curve is the analytical solution, the orange curve is the solution from the evolving-Gaussian experiment, the red curve is the solution from the Gaussian-mean experiment, and the blue curve is the solution from the Gaussian-mode experiment.

We also conduct a set of sensitivity experiments by tuning the standard deviation for the fixed Gaussian in the Gaussian-mean and Gaussian-mode experiments. The results are shown in Figure 5.6(a),(b). Although we can fine-tune the standard deviation of the Gaussian in these ex-

periments to obtain a better solution globally, none of these experiments can fit the curve of the analytical solution exactly. In fact, the analysis x_a is a linear function of the background x_b if using a fixed Gaussian (Gaussian-mean and Gaussian-mode experiment) since:

$$x_a = \left(\frac{\sigma_b^2}{\sigma_o^2 + \sigma_b^2}\right) y_o + \left(\frac{\sigma_o^2}{\sigma_o^2 + \sigma_b^2}\right) x_b \tag{5.13}$$

where σ_o, σ_b, and y_o are constants in these experiments. Since the analytical x_a is nonlinearly related to x_b in this experiment, none of these linear solutions can perfectly fit the analytical solution.

Figure 5.6(c) shows the sensitivity of the evolving-Gaussian method to the number of outer-loop used. We can see that even if we just use one outer-loop, the solutions from the evolving-Gaussian experiment are already better than all of the fixed Gaussian experiments, regardless of how the standard deviation is tuned. With more than two outer-loops, the solutions from the evolving-Gaussian experiments are almost indistinguishable from the analytical solution.

In summary, the evolving-Gaussian method can improve the analysis of the incremental 4D-Var over using a fixed Gaussian error. We will dive deeper into how the evolving-Gaussian method finds the analysis and discuss the conditions where the evolving-Gaussian method is guaranteed to converge to the analytical solution in the next section.

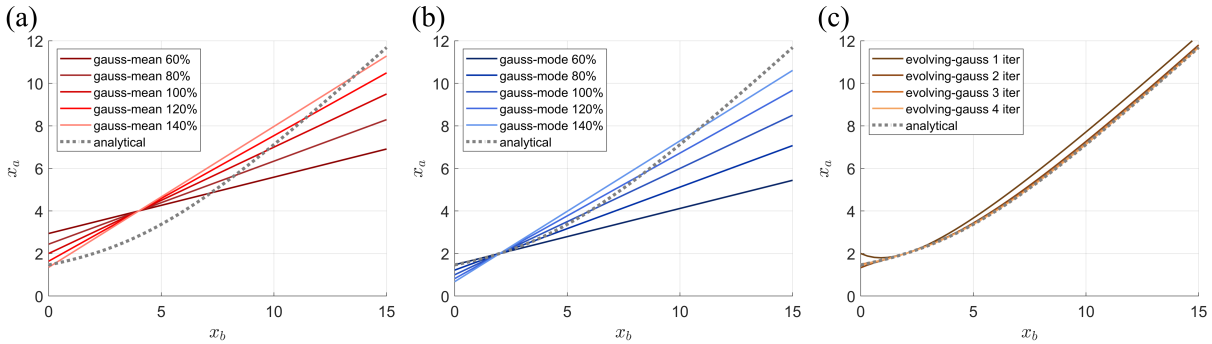

Figure 5.6: Similar to Figure 5.5, but for different sensitivity tests. The different colored curves show the results from (a) the Gaussian-mean experiments with the standard deviation of the Gaussian varied from 60%-140% of the non-Gaussian error (b) the Gaussian-mode experiments with the standard deviation of the Gaussian varied from 60%-140% of the non-Gaussian error (c) the evolving-Gaussian experiments using different number of outer-loop.

5.3.4 Discussion

The idea of the evolving Gaussian method is to generate a sequence of state $\{y^n\}$, and we hope that $\{y^n\}$ converges to y_a. The sequence starts with the background, i.e., $y^0 = y_b$, and proceeds by the following relation:

$$f(y) = \left(\frac{g(y)^2}{g(y)^2 + \sigma_b^2}\right) y_b + \left(\frac{\sigma_b^2}{g(y)^2 + \sigma_b^2}\right) y_o \tag{5.14}$$

where σ_b is the standard deviation of the background error in the observation space and $g(y)$ describes the function of the estimated standard deviation from Equation (5.11) as a function of $y = H(\mathbf{x})$. Note that the sequence is proceeded by $y^{n+1} = f(y^n)$.

To better understand the behavior of $\{y^n\}$, we examine the sequence of the errors $r^n = y^n - y_a$ in Figure 5.7. We can see all the experiments converge towards y_a (i.e., $r^n \to 0$) in 2-3 iterations. We further divide the experiments into two: one with the background y_b larger than the observation y_o (blue curves in Figure 5.7), and the other with $y_b < y_o$ (red curves). We can see that the convergence behavior for the two groups are very different. When $y_b > y_o$, r^n monotonically decreases towards 0. On the other hand, when $y_b < y_o$, r^n oscillates around 0 (i.e., y^n oscillates around y_a) but gradually converges to 0.

To better understand the behavior, we introduce another function for each y_b, the *steady curve* $h(y)$, which is defined between y_o and y_b as:

$$y = \left(\frac{h(y)^2}{h(y)^2 + \sigma_b^2}\right) y_b + \left(\frac{\sigma_b^2}{h(y)^2 + \sigma_b^2}\right) y_o \tag{5.15}$$

The meaning of the steady curve $h(y)$ is the *required* standard deviation of a Gaussian observation error such that the analysis will be y. We can rearrange Equation (5.15) to get

$$h(y) = \sqrt{-\sigma_b^2 \left(\frac{y - y_o}{y - y_b}\right)} \tag{5.16}$$

The steady curves $h(y)$ for $y_b = 0.6$ and $y_b = 8.7$ are shown in the blue curve in Figure 5.8. Note that $h(y_o)$ is always 0, meaning that we need no uncertainty of the observation to obtain an

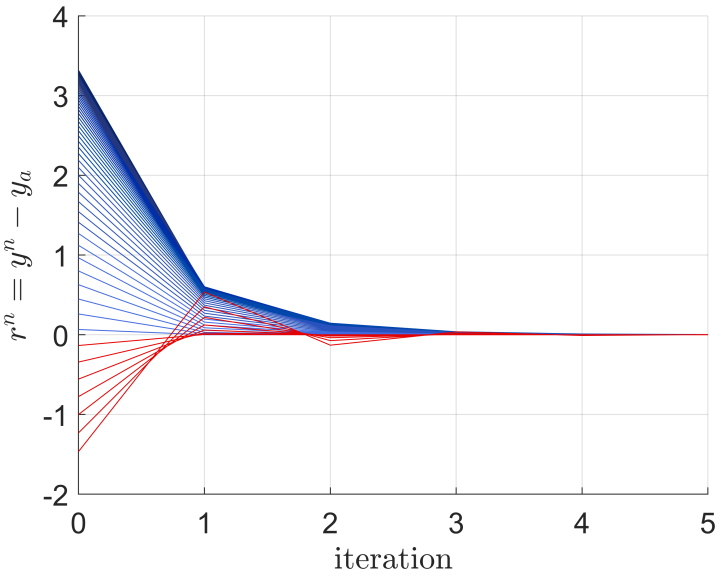

Figure 5.7: The sequence of the errors $r^n = y^n - y_a$ for the evolving-Gaussian experiments starting from different background y_b. The blue curves are the experiments with $y_b > y_o$, and the red curves are $y_b < y_o$.

analysis that is exactly the same as the observed value. $h(y)$ monotonically increases from y_o to y_b, meaning that we trust the background more when the observation uncertainty is larger. The extreme case is when $y = y_b$, where we do not use the information of the observation at all, and we need $h(y_b) = \infty$ to achieve this.

By observing Equation (5.14) and (5.15), we can also write f as a composite function:

$$f(y) = h^{-1}(g(y)) \tag{5.17}$$

With Equation (5.17), we can now better understand how the sequence evolves by f in Figure 5.9. We start with $y^0 = y_b$, and evaluate $g(y^0)$, which is shown by the gray dot on the red curve. Then, we evaluate $h^{-1}(g(y^0))$, for which we go horizontally from the gray dot to the blue curve (the steady curve) and find the corresponding value in the x-axis. We now find y^1 since $y^1 = f(y^0) = h^{-1}(g(y^0))$. We can follow the same procedure to find $y^n = h^{-1}(g(y^{n-1}))$ and so on.

Figure 5.9(a) shows the case where $y_b < y_o$, which corresponds to the red curves in Figure 5.7. Figure 5.9(a) explains why the sequence $\{y^n\}$ oscillates around y_a when $y_b < y_o$. On the other

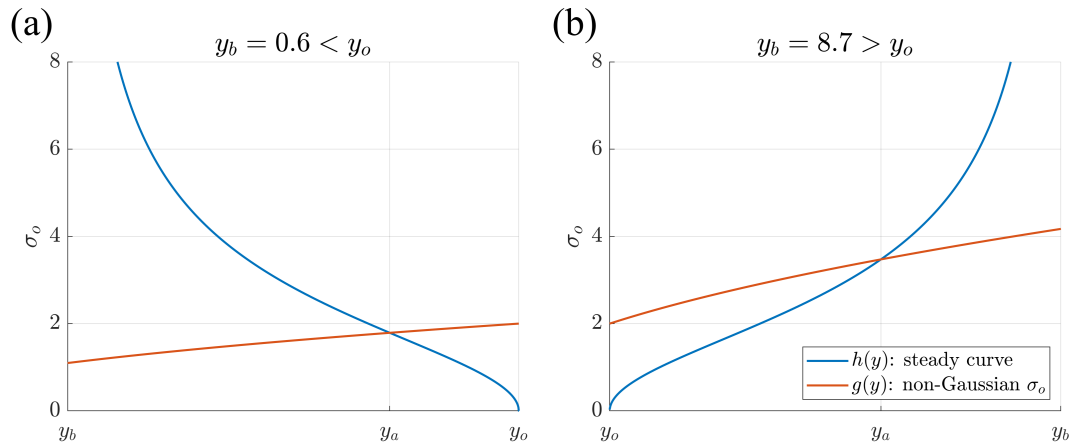

Figure 5.8: Examples of the steady curve $h(y)$ (blue) and the standard deviation $g(y)$ (red) for two cases where (a) the background $y_b = 0.6$ is smaller than the observation $y_o = 2$ (b) $y_b = 8.7$ is larger than $y_o = 2$.

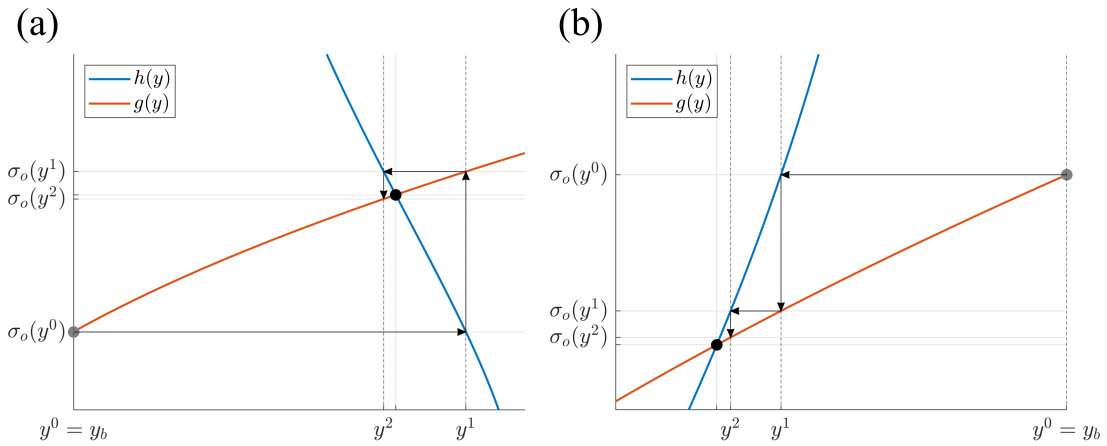

Figure 5.9: Demonstration of how f in Equation (5.17) works to proceed the sequence y^n. The blue curve is $h(y)$ and the red curve is $g(y)$, which are exactly the same as in Figure 5.8. The arrows indicate how the sequence y^n is evolved. The gray dot represents the standard deviation of the Gaussian used in the first outer-loop, and the black dot represents the intersection of $h(y)$ and $g(y)$, so the standard deviation at y_a.

hand, Figure 5.9(b) shows the case where $y_b > y_o$, which corresponds to the blue curves in Figure 5.7. Similarly, Figure 5.9(b) explains why the sequence $\{y^n\}$ monotonically decreases towards y_a when $y_b < y_o$.

We note that the evolving-Gaussian method is not guaranteed to converge to y_a for any arbitrary $g(y)$. One can imagine a tortuous $g(y)$ such that the evolving-Gaussian method will get stuck locally before reaching y_a. Nevertheless, we can still find conditions where the evolving-Gaussian method is guaranteed to converge to y_a when $g(y)$ is monotonic between y_b and y_o:

1. When $g(y)$ increases from y_o to y_b (i.e., when the sign of the slope for $g(y)$ and $h(y)$ is the same, e.g., Figure 5.9(b)), it is required that there exists only one intersection between $h(y)$ and $g(y)$.

2. When $g(y)$ decreases from y_o to y_b (i.e., when the sign of the slope for $g(y)$ and $h(y)$ is different, e.g., Figure 5.9(a)), it is required that the transformation $F = (f \circ f)$ must satisfy

$$if\ y > y_a,\ then\ F(y) < y$$

and

$$if\ y < y_a,\ then\ F(y) > y$$

Please refer to the Appendix D.2 for further details. As a final remark, criterion 1 is a rather easy one to satisfy for most cases. Criterion 2 is a tricky one, but in general this implies a smaller magnitude of the slope of $g(y)$.

5.4 An implementation in the ECMWF-IFS system

We next implement the evolving-Gaussian method for the all-sky satellite microwave radiances in a full-scale operational weather forecasting system, the Integrated Forecasting System (IFS) of the European Centre for Medium-Range Weather Forecasts (ECMWF). The experiment setup is described in Section 5.4.1. We describe the default Gaussian observation error model used in IFS

for the all-sky microwave radiances in Section 5.4.2, and the new non-Gaussian model by using the evolving-Gaussian method in Section 5.4.3. The results from the IFS experiment will be presented in Section 5.4.4.

5.4.1 Experiment setup

In the IFS experiment, we use the 48r1 cycle in IFS and run cycling DA experiments during June 2020 to February 2021. The observation frequency is 12 hr. We will evaluate the effect of the non-Gaussianities by comparing the results from two simulations in IFS: (1) with Gaussian observation error models (default) and (2) non-Gaussian observation error models for selected observations. Specifically, we change the observation error model from Gaussian to non-Gaussian, using the evolving-Gaussian method, for five all-sky microwave channels from the Special Sensor Microwave Imager/Sounder (SSMIS) (Kunkee et al., 2008): ch12 (19-H), ch13 (19-V), ch14 (22-V), ch16 (37-V) and ch17 (91-V). Note that the reasons we change the error models for the all-sky microwave radiances are: (1) They can exhibit non-Gaussian errors (e.g., Bocquet et al., 2010), especially when cloud and precipitation are present. (2) The all-sky satellite radiances can contribute to a significant amount of short-range forecast impact (e.g., Geer et al., 2017). We emphasize that an ideal comparison to see the effect of the non-Gaussianties would be replacing *all* the error models of the observations exhibiting non-Gaussian errors with the evolving-Gaussian method. However, we note that this requires an excessive amount of work and the total effect can be complicated to understand. Therefore, as a demonstration purpose, we only change the observation error model for the five of the microwave channels, and we will focus on its effect on the variables that are sensitive to these five channels.

5.4.2 The default Gaussian error model

The Gaussian error model for the all-sky microwave radiances in the IFS uses the cloud amount dependent error model (Geer and Bauer, 2011). Specifically, it assumes that the observation error is Gaussian but with its variance depending on some predictors (cloud amount here), which accounts for the *heteroscedasticity* of the errors for all-sky microwave radiances. For example, radiative

transfer models can result in larger errors when there is cloud and precipitation, thus leading to a larger observation error variance when there is cloud compared to a clear sky.

The cloud amount dependent error model uses a predictor called symmetric cloud amount ($\overline{C_{37}}$), which is

$$\overline{C_{37}} = \frac{C_{37}^b + C_{37}^o}{2} \tag{5.18}$$

where C_{37}^b and C_{37}^o are the cloud proxy for the background (model simulation) and the observation, respectively. The cloud proxy variable C_{37} is defined based on the normalized polarization difference of the 37 GHz channel:

$$C_{37} = 1 - \frac{T^v - T^h}{T_{clr}^v - T_{clr}^h} \tag{5.19}$$

where T^v and T^h are the vertically and horizontally polarized brightness temperature, and T_{clr}^v and T_{clr}^h are the same brightness temperature but assuming there is no cloud or precipitation. The cloud proxy C_{37} varies between 0 and 1, where 0 represents a clear sky and a value closer to 1 represents an existence of a more opaque cloud.

Geer and Bauer (2011) found that using the symmetric cloud amount $\overline{C_{37}}$ as the predictor makes the distribution of O-B innovation (background departure) more symmetric than using either the background C_{37}^b or the observed C_{37}^o cloud amount. We reproduce similar results using a newer data set collected during September 2015, which is shown in Figure 5.10. Note that the error model in IFS assumes that the uncertainties in the innovation are totally attributed to the observation error. Therefore, Figure 5.10c is used to construct the Gaussian cloud amount dependent error model, with the variable on the y-axis being the predictor. For example, if we would like to know the observation error pdf when the symmetric cloud amount $\overline{C_{37}}$ is 0.5, we can obtain a marginal pdf $p(d, \overline{C_{37}} = 0.5)$ (d is the O-B innovation in x-axis, and $\overline{C_{37}}$ is the symmetric cloud amount in y-axis) by crossing the joint pdf with a horizontal line $y = 0.5$ (symmetric cloud amount $\overline{C_{37}} = 0.5$). We can then estimate the standard deviation of this marginal pdf, which will be used as the standard

deviation of the Gaussian when $\overline{C_{37}} = 0.5$. A summary of the estimated standard deviation as a function of $\overline{C_{37}}$ from Figure 5.10c is shown in Figure 5.11a.

Although using $\overline{C_{37}}$ makes the error distributions more symmetric, careful examinations of these distributions still show some non-Gaussianities. For example, many of the marginal pdfs $p(d, C = c)$ in Figure 5.10c exhibit skewness. Nevertheless, the Gaussian error model parameterizes the conditional pdf in Figure 5.10c with a single value of mean and standard deviation. To compare with the existing Gaussian model, we will use the evolving-Gaussian method to deal with the non-Gaussianities seen here, which will be detailed in the following subsection.

5.4.3 The non-Gaussian error model

In principle, we should first obtain a non-parametric observation error pdf from DOEE and then apply the evolving-Gaussian method to the observation error pdfs for these 5 channels. However, as mentioned in Section 5.4.2, the IFS system uses the innovation pdf as the observation error pdf for the all-sky microwave radiances. Therefore, in order to focus on the effect of the non-Gaussianities, we will also apply the evolving-Gaussian method directly to the innovation pdfs for these channels.

Note that Section 5.2.3 describes the way to estimate the standard deviation $\sigma_o(d)$ as a function of innovation d with a single non-Gaussian pdf. Here, we would like to construct a cloud amount dependent model. The strategy is that we construct several functions $\sigma_o^i(d)$ for some discretized cloud amount i, and interpolate the standard deviation in between. Specifically, we divide the symmetric cloud amount $\overline{C_{37}}$ into 10 bins, i.e., [0.0,0.1], [0.1,0.2], ..., [0.9,1.0] and estimate the pdf in each bin. Then, we estimate $\sigma_o^i(d)$ for the 10 pdfs ($i = 1, ..., 10$). The estimated $\sigma_o^i(d)$ are assumed to be evaluated at $\overline{C_{37}} = 0.05, 0.15, ..., 0.95$ for each bin respectively. In practice, when we need the observation error pdf for e.g., $\overline{C_{37}} = 0.58$, we then interpolate σ_o from $\overline{C_{37}} = 0.55$ and $\overline{C_{37}} = 0.65$. A practical way to estimate $\sigma_o(d)$ from noisy histograms obtained from real data is described in Appendix D.3. As a practical choice, we use 4 parameters to describe each $\sigma_o^i(d)$ as a function of d.

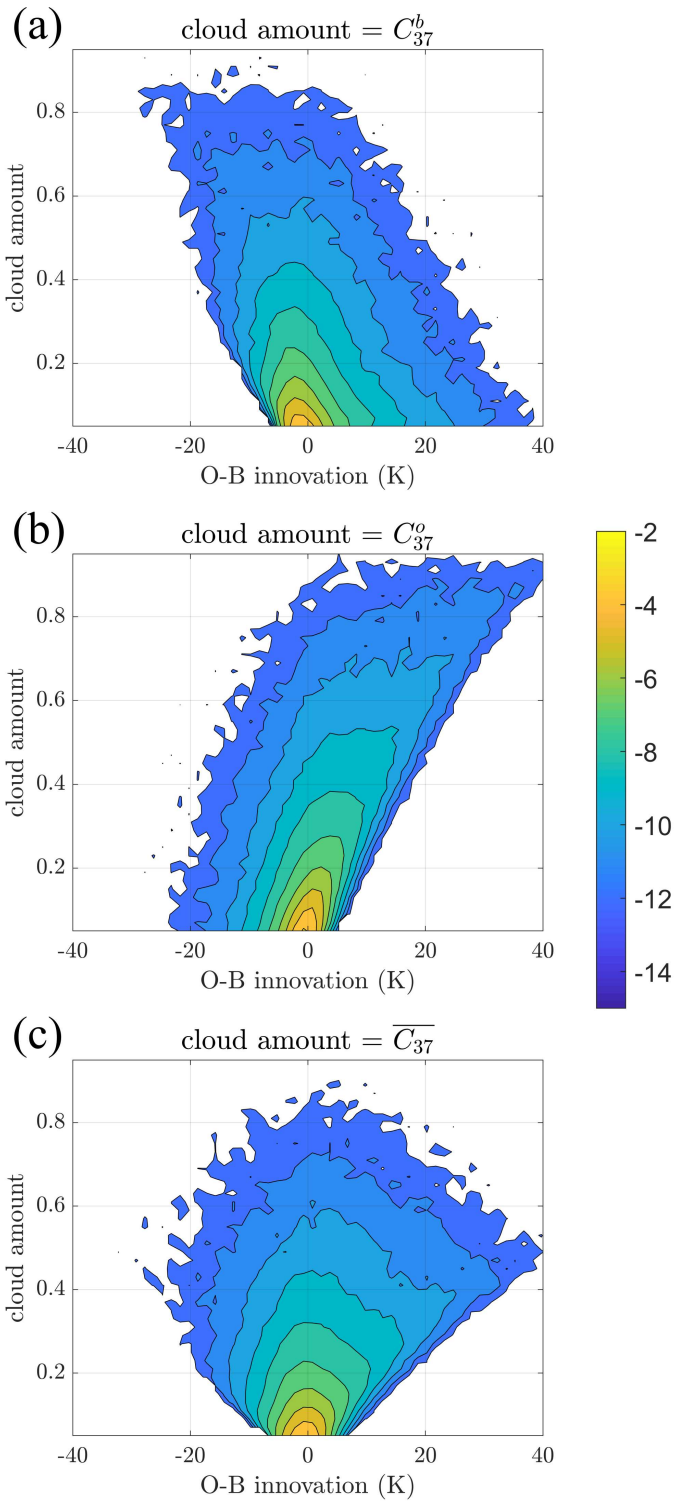

Figure 5.10: The logarithm of the joint distribution of the 19-V innovation (background departure) and cloud amount, where the cloud amount is defined by (a) C_{37}^{b} background cloud amount (b) C_{37}^{o} observed cloud amount (c) $\overline{C_{37}}$ symmetric cloud amount defined in Equation (5.18).

In summary, Figure 5.11 shows an example of the difference between the Gaussian (Section 5.4.2) and non-Gaussian (this Section) cloud amount dependent error model for the horizontally polarized 19 GHz radiances. Note that both models in Fig.5.11 are estimated from the new dataset collected in September 2015. For the Gaussian model (Figure 5.11a), the standard deviation σ_o is only a function of cloud amount $\overline{C_{37}}$, while for the non-Gaussian model (Figure 5.11b) σ_o is both a function of cloud amount and innovation. For the same cloud amount, the σ_o of small innovation d is smaller in the non-Gaussian error model than the Gaussian error model, and the σ_o of large innovation d is larger in the non-Gaussian error model than the Gaussian error model (Figure 5.11c). This suggests that the σ_o in the Gaussian error model is like a smoothed and averaged σ_o of that in the non-Gaussian error model.

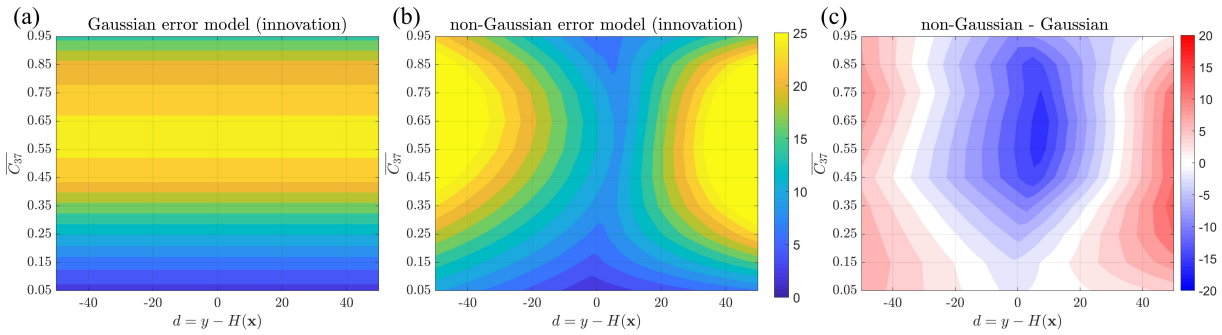

Figure 5.11: The standard deviation σ_o (K) as a function of the symmetric cloud amount $\overline{C_{37}}$ and innovation $d = y - H(\mathbf{x})$ with a (a) Gaussian and (b) non-Gaussian model. (c) shows the difference of the two models, i.e., (b) minus (a).

5.4.4 Results

We compare the results using the Gaussian error model (described in Section 5.4.2) and the non-Gaussian error model (described in Section 5.4.3) here. Note that we do not see significant impact of the non-Gaussian error on the mid-term forecast (i.e., forecast longer than 5 days) (not shown), and this is probably because the five microwave channels are more sensitive to lower-tropospheric humidity, cloud and precipitations, which are more related to the short-term forecast. Therefore, we will focus our comparisons on the short-term, i.e., 12-h forecast in the following.

Since we do not know the truth, it is not trivial to gauge the model forecast performance in the state space. Instead, we compare the forecast with the available observations that have not yet been assimilated. Specifically, we use the O-B diagnostics (observation minus background) collected from the 9-21 h forecast as a metric to evaluate the model performance. The idea is that, if DA can improve the forecast (i.e., the new background for the next DA cycle) by improving the analysis, then the forecast (the new background) becomes closer to the observation. We examine the normalized standard deviation of O-B, which is the O-B standard deviation from the non-Gaussian experiment divided by the O-B standard deviation from the Gaussian experiment. Therefore, a normalized standard deviation smaller than 100% suggests an improvement in the forecast.

Note that the samples of O-B data used in each experiment are the observations that are actively selected for data assimilation (i.e. the observations that pass the data screening, background QC, and varQC in IFS. Observations that are too far from the forecast will not pass these QC). Therefore, the samples in the two experiments are slightly different. In other words, the changes in the O-B standard deviation might come from either the change in the sample, or the change in the quality of the forecasts. Nevertheless, we note that the number of O-B samples for the two experiments are very similar (the difference is mostly smaller than 0.05%). Therefore, we consider the main differences of the O-B standard deviation between two experiments are from the forecast quality.

We first examine the forecast of the all-sky microwave radiances from AMSR2 (Advanced Microwave Scanning Radiometer 2). Six channels (ch7,8,9,10,11,13) are examined, which correspond to similar frequencies of the channels from SSMIS whose observation error models are modified. We note that the observation error model for AMSR2 is not modified, i.e., they are Gaussian in both experiments. Figure 5.12a shows that the normalized standard deviation are reduced for all the six channels globally. We also examine the comparable channels with similar frequencies (ch3,4,5,6,8) from GMI, which indicate very similar results (not shown). Note that although the frequencies that we examine on AMSR2 and GMI are similar to the ones on SSMIS, the observation time and location are different. Therefore, AMSR2 and GMI can be seen

as "semi-independent" observations for the verification purpose. The results from these channels suggest that the forecast of the model variables that are the input to the radiative transfer model for AMSR2 and GMI channels are improved in the non-Gaussian error experiment. These results suggest that there is an improvement in the forecast of lower-tropospheric humidity, cloud and precipitations.

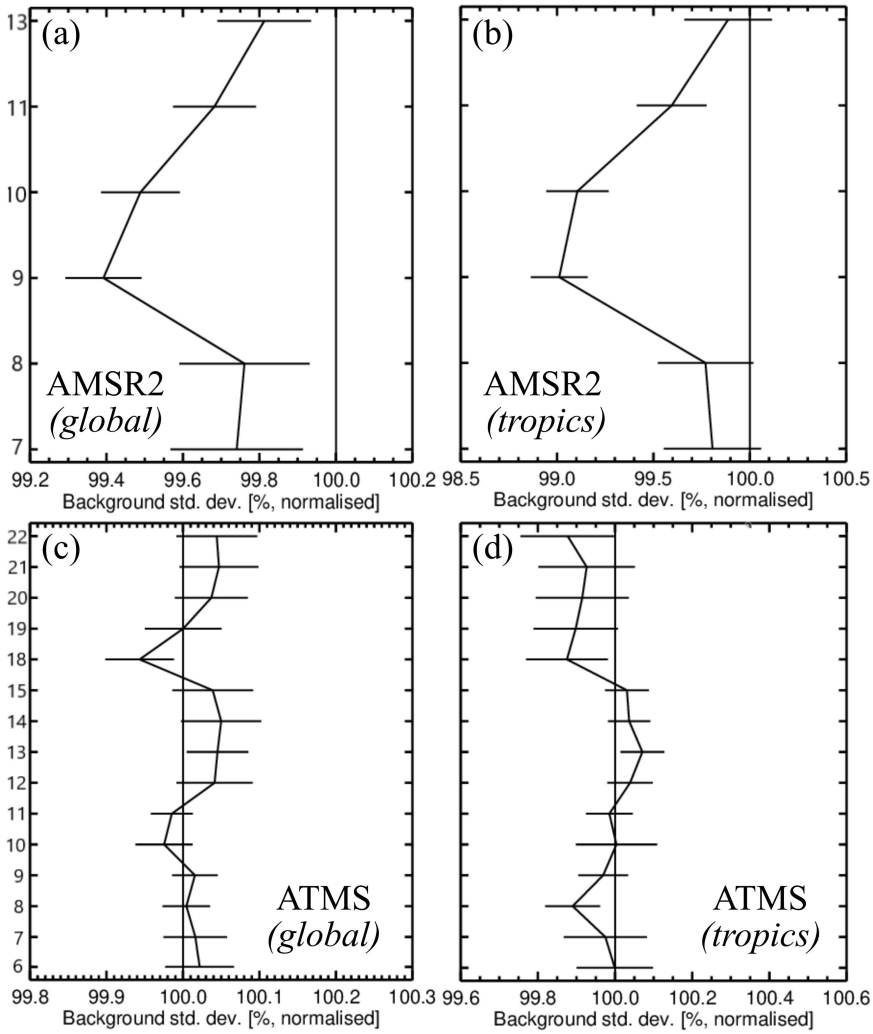

Figure 5.12: The impact of the non-Gaussianities in the observation error model on the forecast of (a)(b) AMSR2 and (c)(d) ATMS microwave observations. The x-axis shows the normalized standard deviation of the 12-h forecast O-B, which is the $stdev(O-B)$ in the non-Gaussian experiment normalized by the $stdev(O-B)$ in the control (Gaussian) experiment. Values smaller than 100% suggests an improvement in the forecast. The y-axis shows the channel numbers that are assimilated into IFS. Confidence bars indicate the 95% confidence interval. (a)(c) show the global statistics and (b)(d) show the statistics only over the tropics.

In addition to the frequencies that are similar to the SSMIS observation (i.e., 19, 22, 37, 91 GHz), we also examine the forecast of the radiances in other frequency range. Figure 5.12c shows the normalized standard deviation from ATMS (Advanced Technology Microwave Sounder). Note that since IFS only assimilate clear-sky radiances for ATMS, only clear-sky observations are verified againt the forecast. Channel 6-15 are temperature channels, which in general shows neutral results with a marginal degradation between ch12-ch15, whose weighting functions peak at upper troposphere (above < 20 hPa). Channel 18-22 are humidity channels, whose weighting functions are shown in Figure 5.13. Figure 5.12c suggests that there is an improvement for ch18 whose weighting function peaks at mid-lower troposphere (Fig.5.13), while a marginal degradation in ch20-22 whose weighting functions peak at upper troposphere. In addition to the microwave channels, we also look at the infrared channels from IASI (Infrared Atmospheric Sounding Interferometer) and CRIS (Cross-track Infrared Sounder). Specifically, examinations of their normalized standard deviation both show improvement for wavenumber around 750-850 cm^{-1} (not shown), which is known as the dirty window and has sensitivity to lower troposphere water vapor. In summary, the results from different satellite sensors and radiances across different frequencies (wavenumbers) indicate that, although there might be marginal degradation of water vapor and temperature in upper-troposphere, the improvement of lower-troposphere humidity is clear.

Besides satellite radiances, we also look at the conventional observations. Figure 5.14 show the composite sounding of observations for humidity, temperature and horizontal wind. The forecast of temperature and the wind are quite neutral. There is a marginal improvement of the humidity sounding, which is similar to the big pictures that we have seen in Figure 5.12. We also check the surface observations, including surface wind and pressure (not shown). The surface wind is overall neutral with some marginal improvement, while there is degradation in the sea surface pressure, especially for the observations from buoys. The normalized standard deviation for the sea surface pressure from buoys shows about 0.2-0.3% degradation. We note that the sea level pressure is an important variable for mid-range forecast. Further investigation is needed to see why the sea surface pressure is deteriorated.

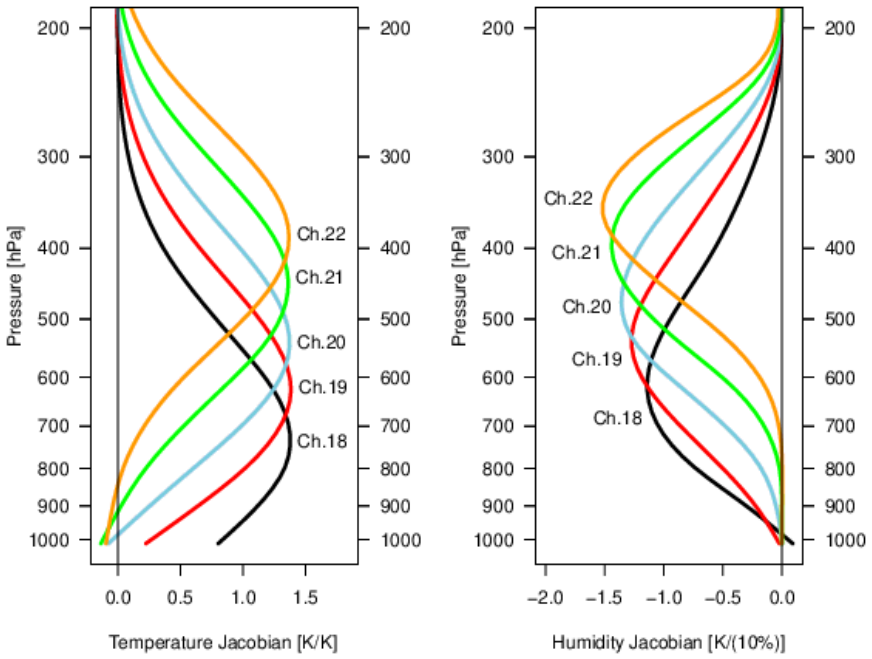

Figure 5.13: The weighting function for humidity channels (ch18-22) on ATMS.

If we restrict the verification only to the observations in the tropics (within 20°S to 20°N), we can observe that the normalized standard deviation becomes smaller compared to the global statistics in general. This is particularly apparent for ch9-10 of AMSR2 (Fig.5.12b), ch19-22 of ATMS (Fig.5.12d), and humidity sounding below 200 hPa (Fig. 5.14b). On the other hand, the temperature sounding (Fig.5.14d) is similar to the global verification, while there seems to be a marginal improvement of the wind sounding below 300 hPa (Fig.5.14f). Besides, the sea surface pressure observation shows a marginal improvement in the tropics (not shown). Specifically, the buoy observation shows a normalized standard deviation of about 0.2% improvement, although the improvement is not statistically significant based on the 95% confidence interval.

To sum up, the verification against observations suggest that the improvement of the short-term forecast of lower troposphere humidity, cloud and precipitations using the non-Gaussian error model is clear. The improvement is more apparent in the tropics. Although the results for other variables are mixed, we emphasize that in this experiment we only change the observation error model for 5 channels of SSMIS, which is a small subset of the whole observing system. In order to comprehensively assess the impact of the non-Gaussianities, it is essential to investigate the

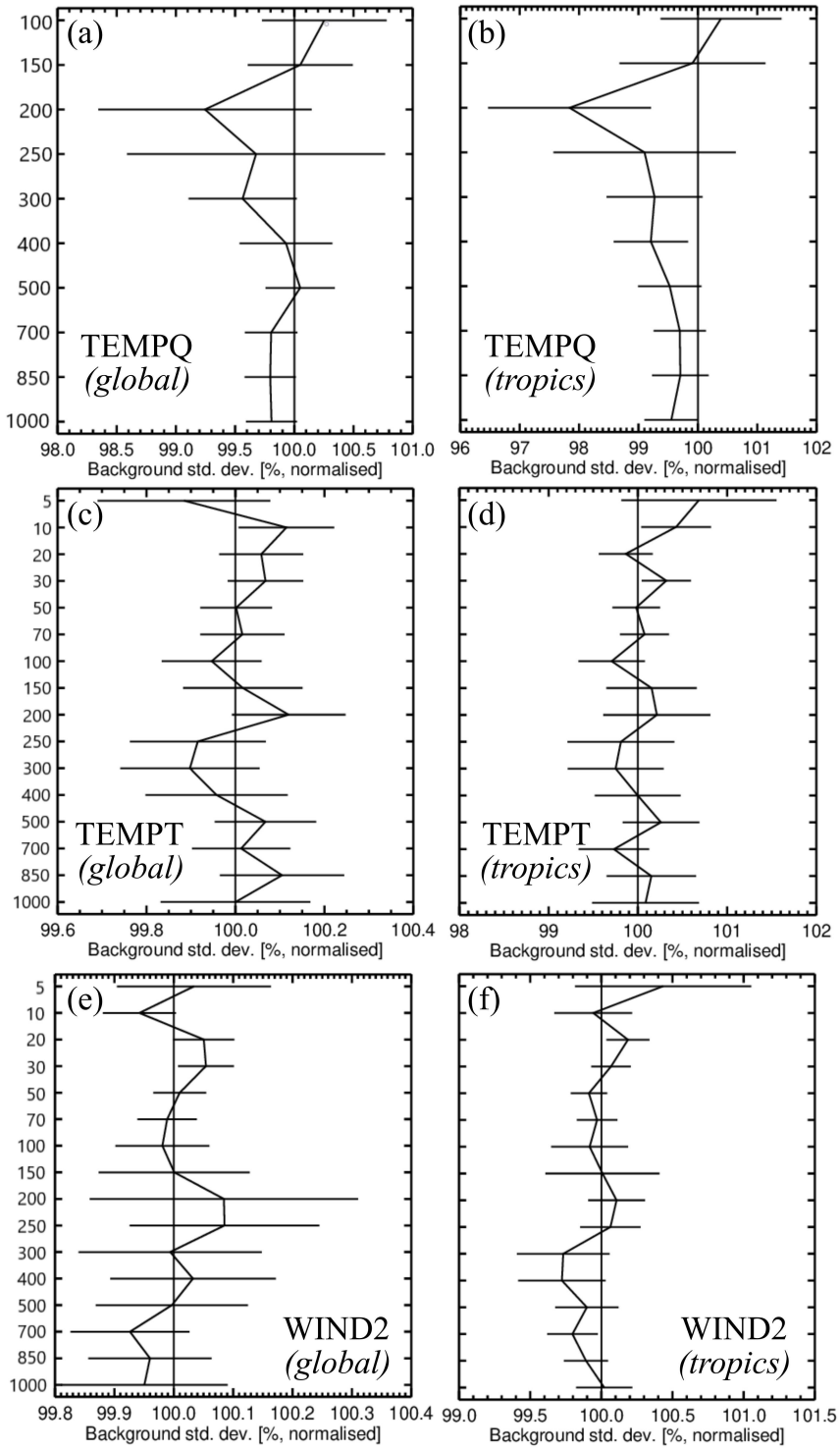

Figure 5.14: Similar to Figure 5.12, but for (a)(b) humidity sounding (c)(d) temperature sounding (e)(f) horizontal wind sounding. The y-axis is pressure in hPa.

potential influence of non-Gaussianities arising from other observations, such as those associated with temperature and wind. This exploration merits further investigation in future research.

5.5 Concluding remarks

We have proposed a new method, the evolving-Gaussian method, to incorporate the non-Gaussianities in the observation errors into the variational method. Specifically, we update the Gaussian likelihood in the observation part of the cost-function J_o in Equation (5.4) in each outer loop to account for the non-Gaussianities. The evolving-Gaussian method does not require re-formulating the cost-function J_o, and therefore its implementation is easy and without additional computational cost.

We first examine the evolving-Gaussian method in an idealized experiment to assimilate the observation with Gamma likelihood. We find that the evolving-Gaussian method outperforms the purely Gaussian methods. Next, we apply the evolving-Gaussian method to the 4D-Var in the the Integrated Forecasting System (IFS) in European Centre for Medium-Range Weather Forecasts (ECMWF). We change the observation error models for 5 channels of the all-sky microwave radiances from Gaussian (default) to non-Gaussian, and compare the 12-h forecast between the two experiments. Based on the O-B diagnostics, the results suggest that the forecast of lower-tropospheric humidity, cloud and precipitations has improved in the non-Gaussian experiment.

Based on the results from the idealized (Section 5.3) and realistic experiments (Section 5.4), the evolving-Gaussian method shows potential to deal with the issues arising from non-Gaussianities in the observation error. Compared to the previously proposed techniques aimed at addressing the non-Gaussianities, one unique aspect of the evolving-Gaussian method is that it is not restricted to a particular parametric distribution. This quality makes it a more versatile approach, which can be used for observations with a general error distribution. Although being more flexible, the evolving-Gaussian method, inevitably, requires more parameters to characterize the error distribution. Note that for Gaussian pdf, we only need 2 parameters (mean and standard deviation), and for Huber norm we need 3 (the two transition points and the standard deviation of the Gaussian in

the center). We note that this is a trade-off problem: the more parameters we use, the more general the distribution we can describe. The number of parameters used in the evolving-Gaussian method to describe the pdf should depend on the how complicated the actual error pdf is, and can be determined by the user. For example, in Section 5.4, we choose to use only 4 parameters to describe the pdf by constructing a piecewise linear fit to the estimated standard deviation (see Appendix D.3 for details), at the cost of losing some details in the pdfs (e.g., see Figure D.3).

One potential application of the evolving-Gaussian method is to also apply it to the background cost-function J_b. Following similar ideas, we could also approximate a non-Gaussian prior using a Gaussian and update the Gaussian in each outer loop. However, we note that choosing an appropriate Gaussian in each outer loop to represent a non-Gaussian prior can be more challenging. This is mainly due to the difficulty in finding a way to properly characterize, especially, the flow-dependent non-Gaussian prior in a high dimensional space. Nevertheless, once we have ways to describe the non-Gaussian prior, in principle we can adopt the idea of evolving-Gaussian method to introduce non-Gaussianities into the prior as well.

Despite holding promises of assimilating observations with non-Gaussian errors, there are still limitations of the evolving-Gaussian method. One limitation is that the evolving-Gaussian method is not always guaranteed to converge to the mode of posterior. The convergence is guaranteed when two criterion for the standard deviation function $g(y)$ (or $\sigma_o(d)$) as described in Section 5.3.4 are met. Nevertheless, we note that these criteria might be too strict (since they are only *sufficient* conditions, but not *necessary* conditions). Moreover, the criteria are derived based on the assumption of linear observation operator, and that the inner loops find the exact solution that minimizes each quadratic function. In practice, the linearization of the nonlinear observation operator might not be perfect and these assumptions might not be exact. Still, one possible solution to improve the evolving-Gaussian method is to directly find the intersection between $h(y)$ and $g(y)$ in Figure 5.9. This finds the mode of the posterior in one outer loop if H is linear. However, this will add computational cost since we need to find the intersection between $h(y)$ and $g(y)$ for each observation. In addition, this makes the gradient of the cost-function different from the exact one,

whose effect to the 4D-Var is less clear. On the other hand, another possible way to improve the evolving-Gaussian method is to also replace the Gaussian in each inner loop. This ensures that during every step of the minimization process the gradient of J_o is exact. However, as discussed in Section 5.2.3, this makes the cost-function non-quadratic, and it is not clear whether it will require more inner loops to reach convergence by doing so. Finally, as stated in Bonavita et al. (2018), there is no guarantee of global convergence of the incremental 4D-Var algorithm which is based on the Gauss-Newton method. In other word, the guarantee of convergence to the mode of posterior is probably a more fundamental issue in 4D-Var. At least, the evolving-Gaussian method offers a practical approach to incorporate information from the non-Gaussianities, and the initial findings do show promise. This indicates the need for further exploration of the evolving-Gaussian method's complete potential, through assimilating a larger variety of observations and/or comparing with other non-Gaussian techniques based on a assumed parametric form in the future.

Chapter 6

Conclusions

In this final chapter, we will first summarize the dissertation in Section 6.1 by looking back at the research questions proposed in Chapter 1, answering these questions, and identifying remaining issues. In Section 6.2, we will then discuss the significance, implications, and future outlook regarding the findings in this dissertation.

6.1 Look back at the research questions

Research Question 1

> *Is there a method that can efficiently draw samples from a non-Gaussian posterior pdf in a high-dimensional system?*

In Chapter 2, we looked into a new development of particle filters, the particle flow filter (PFF), that avoids the weight degeneracy issue in the standard particle filters even with only a small number of particles. Therefore, the PFF has the potential to solve the fully non-linear and non-Gaussian DA problem in a high-dimensional system. Although promising, PFF had not been fully tested in a high-dimensional system, except for a 40-dimensional system.

Therefore, in this work we implement PFF in a 1000-dimensional Lorenz-96 model. We setup the experiment by observing 25% of the system with only 20 particles, which mimics an atmospheric problem. We find that in this "atmospheric-like" setup, the kernel in PFF plays a crucial role. The key contribution of this work is that we propose to explore matrix-valued kernels instead of scalar kernels, in order to prevent the collapse of particles in those directions that are observed. We also demonstrate that PFF can better assimilate non-linear observations over an LETKF (Local Ensemble Transform Kalman Filter). In addition, the PFF can also represent a multi-modal posterior pdf. In short, we present the ability of PFF to solve the non-linear and non-Gaussian DA problem in a high-dimensional system with a setup similar to an atmospheric DA problem.

There are two remaining issues. First, although we try to setup an experiment that mimics the atmospheric problem, the real problem is still different and more complicated, making it difficult to generalize the results. Nevertheless, the work done in Chapter 2 is a necessary step for evaluating a new DA method, and we do demonstrate some unique strengths of the PFF. Second, the algorithm we use in Chapter 2 is a relatively naïve algorithm, which does not scale well with dimension. In order to implement the PFF in an atmospheric model, it is necessary to develop a scalable algorithm, especially one that can run in parallel.

Research Question 2

> *Following research question 1, can we use this method to sample a non-Gaussian posterior pdf in an atmospheric problem?*

In Chapter 3, we devise a parallel algorithm for the PFF, called PFF-DART, in a geoscience data assimilation community software, Data Assimilation Research Testbed (DART), to implement the PFF in atmospheric models. Note that PFF-DART is mathematically different from the original PFF. This is because the original PFF does not fit into the parallel coding structure in DART. Nevertheless, we have shown that PFF-DART can be derived from the generalized Two-Step Ensemble Filtering (TSEF) framework. PFF-DART can update each model variable independently from each observation, which allows PFF-DART to run in parallel.

Although PFF-DART is different from the original PFF, we demonstrate that PFF-DART retains the ability of the PFF to assimilate non-linear and non-Gaussian observations. Specifically, we conduct a year-long cycling observing system simulation experiment with an atmospheric model. We show that PFF-DART is able to produce stable and comparable results to the EAKF for linear and Gaussian observations, and improved results for nonlinear and non-Gaussian observations over existing DA methods. We note that this is the first time that we show that implementing the PFF for atmospheric problem is certainly possible.

There are some remaining issues. One is that the current PFF-DART algorithm is based on a Gaussian prior. Nevertheless, we note that a parametric prior is not needed for PFF-DART, as long as we can derive the gradient of the log of prior $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ at each particles' location. There

are many ways to estimate $\nabla_{\mathbf{x}} \log p(\mathbf{x})$, and they are indeed possible to be incorporated into PFF-DART in the future. Second, the atmospheric model that we use in this chapter is rather simple, which does not include, e.g., microphysical processes that can make the errors more non-Gaussian. Therefore, the next step is to further examine PFF-DART in a model that can resolve convective-scale processes, e.g., Weather Research and Forecasting (WRF) model. Lastly, the current PFF-DART (30 iterations) takes about 6 times longer to compute than the EAKF. Nevertheless, there are still some redundancies in the code due to some technical difficulties. We can expect a more efficient PFF-DART if we can optimize the code better. In addition, in the future we would like to explore ways to reduce the number of iterations used in the PFF-DART, e.g., by exploring proposal densities.

Research Question 3

> *Is it possible to estimate the full distribution of the observation error ε^o even when it is non-Gaussian? What are the distributions of the observation errors for the all-sky satellite microwave radiances (which form an important part of the global observing system)? When do these observations exhibit non-Gaussian errors?*

In Chapter 4, we develop a method, called Deconvolution-based Observation Error Estimation (DOEE), which is the first method that can estimate a full non-parametric pdf of the observation error ε^o. DOEE takes its foundation from the background subtraction method, but relaxes two assumptions, namely, the linear and Gaussian assumptions in the methodology.

We show in an idealized experiment that DOEE can sucessfully retrieve a skewed, or multi-modal pdf of ε^o. Then, we implement DOEE for the observations from the all-sky microwave radiances and construct an observation error model that depends on the cloud amount. We can identify many non-Gaussian features in ε^o, especially for the observations with large total column water vapor.

The remaining question for this work is: how are we going to use the information of non-Gaussian ε^o in DA? One possible solution is to use PFF or any other non-parametric filter, which does not rely on Gaussian assumptions for ε^o. However, we note that most of the operational

weather forecasting centers still use DA methods that are based on Gaussian ε^o, e.g., the variational methods. Therefore, this brings about the last research question in this dissertation.

Research Question 4

How do we incorporate non-Gaussian observation errors (e.g., from the outcome of research question 3) into the variational method in general? How do the non-Gaussianities affect the accuracy of forecasts?

In Chapter 5, we propose a method, called evolving-Gaussian method, as a general way to incorporate non-Gaussian ε^o into a variational method. We first conduct a series of idealized experiments assimilating observations with non-Gaussian ε^o. These experiments show that 4D-Var with the evolving-Gaussian method can better identify the mode of the non-Gaussian posterior pdf over the default 4D-Var assuming Gaussian ε^o.

One advantage of the evolving-Gaussian method is that it does not require a reformulation of the cost-function in the variational method. Therefore, implementing the evolving-Gaussian method immediately into a full-scale operational weather forecasting model is possible. We test the performance of the 4D-Var with evolving-Gaussian method in the Integrated Forecasting System (IFS) in European Centre for Medium-Range Weather Forecasts (ECMWF). We find that by replacing the observation error model of the all-sky microwave radiances from Gaussian to non-Gaussian, the short-term forecast of the cloud, precipitations and water vapor is improved. Particularly, these improvements are more prominent in the tropics.

Note that in the IFS experiment, we only substitute the observation error model for 5 selected microwave channels. There are a lot of observations, even for in-situ conventional observations, that can exhibit non-Gaussian ε^o. More significant forecast impact can be expected when the non-Gaussianities from ε^o for more observation types are treated explicitly in the future.

6.2 Future outlook and implications

Finding a unified, consistent, and perfect solution that can effectively address all the issues associated with non-Gaussianities and non-linearities in weather forecasting is challenging and

has not been achieved yet. While advanced DA methods may theoretically provide solutions to the problem, practical implementation is constrained by the limitations imposed by finite computational resources. Despite the computational constraints, the particle flow filter (PFF) is still a promising candidate for the solution for several reasons. For one reason, the PFF is a flexible method that can adapt to many different situations. For example, although the PFF is not immune to the impact of sampling errors like any other non-parametric filter, the PFF can still assume a parameteric distribution like Gaussian, or even non-Gaussian, when the number of particles is considered to be too small to properly depict the distributions. Moreover, the PFF is constructed in a similar way to variational methods that are commonly used in many operational centers. Although the PFF is explored as a filter in this dissertation, it can be extended to a particle flow smoother (PFS) [4]. A PFS is similar to "an ensemble of 4D-Vars", promoted by e.g. ECMWF. The difference is that "an ensemble of 4D-Vars" does not solve the fully nonlinear problem, while a PFS, through communications between particles at each iteration during the minimization, does solve the fully nonlinear problem. As previously mentioned, instead of making one drastic transition towards a fully non-linear and non-Gaussian approach, operational centers favor incremental changes that gradually add non-linearities and non-Gaussianities into DA. This gradual approach involves introducing these elements from different aspects, allowing for a more controlled and manageable progression towards more advanced DA. Given the quality of PFF as described above, PFF, or PFS, can be a useful method for these transitions.

In addition, the observation error plays a crucial role in DA. The Deconvolution-based Observation Error Estimation (DOEE) provides a way to directly estimate a non-parametric observation error pdf for the first time. Some may argue that replacing a Gaussian with a fully non-parametric observation error pdf in DA is a big change, while DOEE can also be used as a guidance to identify the parametric form for the observation error. We emphasize, again, that the observation error in

[4]The difference between the PFF and PFS is as follows: denote the posterior pdf in the PFF as $p(\mathbf{x}_t|\mathbf{y}_t)$, where the subscript refers to the time index. We extend the model state $\mathbf{x}_t$ to be the model states within a time window $\mathbf{x}_{1:t}$ in the PFS, i.e., the posterior pdf becomes $p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})$. This means that a state at any time in the window will be updated by all observations in the window, including those in its future. When we only have one particle, the PFF is reduced to 3D-Var, and the PFS is reduced to 4D-Var.

DA is not only associated with the observation itself, but is largely associated with the errors in the observation operator (i.e., the representation error). No observation operator is perfect, and we should know their errors. We would like to argue that a proper estimate of the observation error pdf can be as important as an accurate observation operator in DA. Besides, with the development of the machine learning (ML), there have been suggestions to replace computationally expensive observation operators, e.g., one involves the radiative transfer calculations, with surrogate models based on ML. Direct estimation of the representation error for ML-based surrogate observation operator can be challenging. Nevertheless, in this case, DOEE can be used to estimate the total observation error associated with the ML-based surrogate observation operator.

As a final remark, we highlight that with the advancement of weather forecasting models, which aim to resolve more subgrid and convective processes, along with the increasing availability of remotely-sensed observations, the need for non-linear and non-Gaussian DA methods is becoming indispensable for weather forecasting. Furthermore, these methods will play an increasingly crucial role, as they allow the effective incorporation of the observations that were previously underused or even discarded by the linear and Gaussian DA methods. We acknowledge that we have not solved the full problem in this dissertation yet, while we provide important guidance that can be the foundation for the future non-linear and non-Gaussian DA research, for weather forecasting but also for many other forecasting problems in the geosciences and beyond.

References

Ades, M., and P. J. Van Leeuwen, 2015: The equivalent-weights particle filter in a high-dimensional system. *Quarterly Journal of the Royal Meteorological Society*, **141 (687)**, 484–503.

Anderson, E., and H. Järvinen, 1999: Variational quality control. *Quarterly Journal of the Royal Meteorological Society*, **125 (554)**, 697–722.

Anderson, J., T. Hoar, K. Raeder, H. Liu, N. Collins, R. Torn, and A. Avellano, 2009: The data assimilation research testbed: A community facility. *Bulletin of the American Meteorological Society*, **90 (9)**, 1283–1296.

Anderson, J. L., 2001: An ensemble adjustment kalman filter for data assimilation. *Monthly weather review*, **129 (12)**, 2884–2903.

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

Anderson, J. L., 2012: Localization and sampling error correction in ensemble kalman filter data assimilation. *Monthly Weather Review*, **140 (7)**, 2359–2371.

Anderson, J. L., 2022: A quantile-conserving ensemble filter framework. part i: Updating an observed variable. *Monthly Weather Review*, **150 (5)**, 1061–1074.

Anderson, J. L., 2023: A quantile-conserving ensemble filter framework. part ii: Regression of observation increments in a probit and probability integral transformed space. *Monthly Weather Review*, **In review**.

Anderson, J. L., and Coauthors, 2004: The new gfdl global atmosphere and land model am2–lm2: Evaluation with prescribed sst simulations. *Journal of Climate*, **17 (24)**, 4641–4673.

Andersson, E., 2003: Modelling the temporal evolution of innovation statistics. *This volume*, 153–164.

Anthes, R., and T. Rieckh, 2018: Estimating observation and model error variances<? xmltex\break?> using multiple data sets. *Atmospheric Measurement Techniques*, **11 (7)**, 4239–4260.

Arakawa, A., and V. R. Lamb, 1977: Computational design of the basic dynamical processes of the ucla general circulation model. *General circulation models of the atmosphere*, **17 (Supplement C)**, 173–265.

Ba, J., M. A. Erdogdu, M. Ghassemi, S. Sun, T. Suzuki, D. Wu, and T. Zhang, 2021: Understanding the variance collapse of svgd in high dimensions. *International Conference on Learning Representations*.

Bauer, P., A. J. Geer, P. Lopez, and D. Salmond, 2010: Direct 4d-var assimilation of all-sky radiances. part i: Implementation. *Quarterly Journal of the Royal Meteorological Society*, **136 (652)**, 1868–1885.

Bengtsson, T., C. Snyder, and D. Nychka, 2003: Toward a nonlinear ensemble filter for high-dimensional systems. *Journal of Geophysical Research: Atmospheres*, **108 (D24)**.

Bishop, C. H., 2016: The gigg-enkf: ensemble kalman filtering for highly skewed non-negative uncertainty distributions. *Quarterly Journal of the Royal Meteorological Society*, **142 (696)**, 1395–1412.

Bishop, C. H., 2019: Data assimilation strategies for state-dependent observation error variances. *Quarterly Journal of the Royal Meteorological Society*, **145 (718)**, 217–227.

Bishop, C. H., B. J. Etherton, and S. J. Majumdar, 2001: Adaptive sampling with the ensemble transform kalman filter. part i: Theoretical aspects. *Monthly weather review*, **129 (3)**, 420–436.

Bocquet, M., C. A. Pires, and L. Wu, 2010: Beyond gaussian statistical modeling in geophysical data assimilation. *Monthly Weather Review*, **138 (8)**, 2997–3023.

Bonavita, M., A. J. Geer, and M. Hamrud, 2020: All-sky microwave radiances assimilated with an ensemble kalman filter. *Monthly Weather Review*, **148 (7)**, 2737–2760.

Bonavita, M., P. Lean, and E. Holm, 2018: Nonlinear effects in 4d-var. *Nonlinear Processes in Geophysics*, **25 (3)**, 713–729.

Bormann, N., and P. Bauer, 2010: Estimates of spatial and interchannel observation-error characteristics for current sounder radiances for numerical weather prediction. i: Methods and application to atovs data. *Quarterly Journal of the Royal Meteorological Society*, **136 (649)**, 1036–1050.

Bormann, N., M. Bonavita, R. Dragani, R. Eresmaa, M. Matricardi, and A. McNally, 2016: Enhancing the impact of iasi observations through an updated observation-error covariance matrix. *Quarterly Journal of the Royal Meteorological Society*, **142 (697)**, 1767–1780.

Bretherton, C. S., M. E. Peters, and L. E. Back, 2004: Relationships between water vapor path and precipitation over the tropical oceans. *Journal of climate*, **17 (7)**, 1517–1528.

Bringi, V. N., and V. Chandrasekar, 2001: *Polarimetric Doppler weather radar: principles and applications*. Cambridge university press.

Burgers, G., P. J. Van Leeuwen, and G. Evensen, 1998: Analysis scheme in the ensemble kalman filter. *Monthly weather review*, **126 (6)**, 1719–1724.

Chambon, P., S. Q. Zhang, A. Y. Hou, M. Zupanski, and S. Cheung, 2014: Assessing the impact of pre-gpm microwave precipitation observations in the goddard wrf ensemble data assimilation system. *Quarterly Journal of the Royal Meteorological Society*, **140 (681)**, 1219–1235.

Chan, M.-Y., J. L. Anderson, and X. Chen, 2020: An efficient bi-gaussian ensemble kalman filter for satellite infrared radiance data assimilation. *Monthly Weather Review*, **148 (12)**, 5087–5104.

Chorin, A. J., and X. Tu, 2009: Interpolation and iteration for nonlinear filters. *arXiv preprint arXiv:0910.3241*.

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

DAReS, 2022: The data assimilation research testbed (version 10.6.5) [software]. Boulder, Colorado: UCAR/NCAR/CISL/DAReS, http://doi.org/10.5065/D6WQ0202.

Daum, F., and J. Huang, 2011: Particle flow for nonlinear filters. *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 5920–5923.

Desroziers, G., L. Berre, B. Chapnik, and P. Poli, 2005: Diagnosis of observation, background and analysis-error statistics in observation space. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, **131 (613)**, 3385–3396.

Evensen, G., 1994: Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, **99 (C5)**, 10 143–10 162.

Evensen, G., 2018: Analysis of iterative ensemble smoothers for solving inverse problems. *Computational Geosciences*, **22 (3)**, 885–908.

Fletcher, S., 2010: Mixed gaussian-lognormal four-dimensional data assimilation. *Tellus A: Dynamic Meteorology and Oceanography*, **62 (3)**, 266–287.

Fletcher, S., and M. Zupanski, 2006a: A data assimilation method for log-normally distributed observational errors. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, **132 (621)**, 2505–2519.

Fletcher, S. J., and M. Zupanski, 2006b: A hybrid multivariate normal and lognormal distribution for data assimilation. *Atmospheric Science Letters*, **7 (2)**, 43–46.

Freedman, D., and P. Diaconis, 1981: On the histogram as a density estimator: L 2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, **57 (4)**, 453–476.

Gaspari, G., and S. E. Cohn, 1999: Construction of correlation functions in two and three dimensions. *Quarterly Journal of the Royal Meteorological Society*, **125 (554)**, 723–757.

Geer, A., and Coauthors, 2017: The growing impact of satellite observations sensitive to humidity, cloud and precipitation. *Quarterly Journal of the Royal Meteorological Society*, **143 (709)**, 3189–3206.

Geer, A. J., and P. Bauer, 2011: Observation errors in all-sky data assimilation. *Quarterly Journal of the Royal Meteorological Society*, **137 (661)**, 2024–2037.

Geer, A. J., and Coauthors, 2018: All-sky satellite data assimilation at operational weather forecasting centres. *Quarterly Journal of the Royal Meteorological Society*, **144 (713)**, 1191–1217.

Goodliff, M. R., S. J. Fletcher, A. J. Kliewer, A. S. Jones, and J. M. Forsythe, 2022: Non-gaussian detection using machine learning with data assimilation applications. *Earth and Space Science*, **9 (4)**, e2021EA001 908.

Grooms, I., 2022: A comparison of nonlinear extensions to the ensemble kalman filter: Gaussian anamorphosis and two-step ensemble filters. *Computational Geosciences*, **26 (3)**, 633–650.

Grubbs, F. E., 1948: On estimating precision of measuring instruments and product variability. *Journal of the American Statistical Association*, **43 (242)**, 243–264.

Hamill, T. M., 2001: Interpretation of rank histograms for verifying ensemble forecasts. *Monthly Weather Review*, **129 (3)**, 550–560.

Hodyss, D., and N. Nichols, 2015: The error of representation: Basic understanding. *Tellus A: Dynamic Meteorology and Oceanography*, **67 (1)**, 24 822.

Houtekamer, P. L., and H. L. Mitchell, 1998: Data assimilation using an ensemble kalman filter technique. *Monthly Weather Review*, **126 (3)**, 796–811.

Hu, C.-C., and P. J. van Leeuwen, 2021: A particle flow filter for high-dimensional system applications. *Quarterly Journal of the Royal Meteorological Society*, **147 (737)**, 2352–2374.

Huber, P. J., 1972: The 1972 wald lecture robust statistics: A review. *The Annals of Mathematical Statistics*, **43 (4)**, 1041–1067.

Hunt, B. R., E. J. Kostelich, and I. Szunyogh, 2007: Efficient data assimilation for spatiotemporal chaos: A local ensemble transform kalman filter. *Physica D: Nonlinear Phenomena*, **230 (1-2)**, 112–126.

Janjić, T., and Coauthors, 2018: On the representation error in data assimilation. *Quarterly Journal of the Royal Meteorological Society*, **144 (713)**, 1257–1278.

Kunkee, D. B., G. A. Poe, D. J. Boucher, S. D. Swadley, Y. Hong, J. E. Wessel, and E. A. Uliana, 2008: Design and evaluation of the first special sensor microwave imager/sounder. *IEEE Transactions on Geoscience and Remote sensing*, **46 (4)**, 863–883.

Kurosawa, K., and J. Poterjoy, 2023: A statistical hypothesis testing strategy for adaptively blending particle filters and ensemble kalman filters for data assimilation. *Monthly Weather Review*, **151 (1)**, 105–125.

Leviyev, A., J. Chen, Y. Wang, O. Ghattas, and A. Zimmerman, 2022: A stochastic stein variational newton method. *arXiv preprint arXiv:2204.09039*.

Lien, G.-Y., E. Kalnay, and T. Miyoshi, 2013: Effective assimilation of global precipitation: Simulation experiments. *Tellus A: Dynamic Meteorology and Oceanography*, **65 (1)**, 19 915.

Liu, Q., and D. Wang, 2016: Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, **29**.

Lonitz, K., and A. J. Geer, 2020: Reducing the drying effect through a water vapour correction to the all-sky error model. *EUMETSAT/ECMWF Fellowship Programme Research Reports*, **53**, 25.

Lorenc, A., 1984: Analysis methods for the quality control of observations. *Workshop on the Use and Quality Control of Meteorological Observations (ECMWF: 6-9 november 1984)*, 397–428.

Lorenz, E. N., and K. A. Emanuel, 1998: Optimal sites for supplementary weather observations: Simulation with a small model. *Journal of the Atmospheric Sciences*, **55 (3)**, 399–414.

Lorenz, E. N., and K. Haman, 1996: The essence of chaos. *Pure and Applied Geophysics*, **147 (3)**, 598–599.

Lu, J., Y. Lu, and J. Nolen, 2019: Scaling limit of the stein variational gradient descent: The mean field regime. *SIAM Journal on Mathematical Analysis*, **51 (2)**, 648–671.

Morzfeld, M., X. Tu, E. Atkins, and A. J. Chorin, 2012: A random map implementation of implicit filters. *Journal of Computational Physics*, **231 (4)**, 2049–2066.

Nocedal, J., and S. Wright, 2006: Springer series in operations research and financial engineering. *Numerical optimization*, Springer New York.

Pires, C. A., O. Talagrand, and M. Bocquet, 2010: Diagnosis and impacts of non-gaussianity of innovations in data assimilation. *Physica D: Nonlinear Phenomena*, **239 (17)**, 1701–1717.

Posselt, D. J., D. Hodyss, and C. H. Bishop, 2014: Errors in ensemble kalman smoother estimates of cloud microphysical parameters. *Monthly Weather Review*, **142 (4)**, 1631–1654.

Poterjoy, J., 2016: A localized particle filter for high-dimensional nonlinear systems. *Monthly Weather Review*, **144 (1)**, 59–76.

Poterjoy, J., L. Wicker, and M. Buehner, 2019: Progress toward the application of a localized particle filter for numerical weather prediction. *Monthly Weather Review*, **147 (4)**, 1107–1126.

Pulido, M., and P. J. van Leeuwen, 2019: Sequential monte carlo with kernel embedded mappings: The mapping particle filter. *Journal of Computational Physics*, **396**, 400–415.

Pulido, M., P. J. vanLeeuwen, and D. J. Posselt, 2019: Kernel embedded nonlinear observational mappings in the variational mapping particle filter. *Computational Science–ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part IV 19*, Springer, 141–155.

Rieckh, T., and R. Anthes, 2018: Evaluating two methods of estimating error variances using simulated data sets with known errors. *Atmospheric Measurement Techniques*, **11 (7)**, 4309–4325.

Schiro, K. A., J. D. Neelin, D. K. Adams, and B. R. Lintner, 2016: Deep convection and column water vapor over tropical land versus tropical ocean: A comparison between the amazon and the tropical western pacific. *Journal of the Atmospheric Sciences*, **73 (10)**, 4043–4063.

Semane, N., R. Anthes, J. Sjoberg, S. Healy, and B. Ruston, 2022: Comparison of desroziers and three-cornered hat methods for estimating cosmic-2 bending angle uncertainties. *Journal of Atmospheric and Oceanic Technology*, **39 (7)**, 929–939.

Sjoberg, J. P., R. A. Anthes, and T. Rieckh, 2021: The three-cornered hat method for estimating error variances of three or more atmospheric datasets. part i: Overview and evaluation. *Journal of Atmospheric and Oceanic Technology*, **38 (3)**, 555–572.

Snyder, C., T. Bengtsson, P. Bickel, and J. Anderson, 2008: Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, **136 (12)**, 4629–4640.

Tavolato, C., and L. Isaksen, 2015: On the use of a huber norm for observation quality control in the ecmwf 4d-var. *Quarterly Journal of the Royal Meteorological Society*, **141 (690)**, 1514–1527.

Tippett, M. K., J. L. Anderson, C. H. Bishop, T. M. Hamill, and J. S. Whitaker, 2003: Ensemble square root filters. *Monthly weather review*, **131 (7)**, 1485–1490.

van Leeuwen, P. J., 2003: Nonlinear ensemble data assimilation for the ocean. *Seminar on recent developments in data assimilation for atmosphere and ocean, ECMWF*.

van Leeuwen, P. J., 2009: Particle filtering in geophysical systems. *Monthly Weather Review*, **137 (12)**, 4089–4114.

van Leeuwen, P. J., 2010: Nonlinear data assimilation in geosciences: an extremely efficient particle filter. *Quarterly Journal of the Royal Meteorological Society*, **136 (653)**, 1991–1999.

Van Leeuwen, P. J., 2015: Representation errors and retrievals in linear and nonlinear data assimilation. *Quarterly Journal of the Royal Meteorological Society*, **141 (690)**, 1612–1623.

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 (731)**, 2815–2825.

van Leeuwen, P. J., H. R. Künsch, L. Nerger, R. Potthast, and S. Reich, 2019: Particle filters for high-dimensional geoscience applications: A review. *Quarterly Journal of the Royal Meteorological Society*, **145 (723)**, 2335–2365.

Whitaker, J. S., and T. M. Hamill, 2002: Ensemble data assimilation without perturbed observations. *Monthly weather review*, **130 (7)**, 1913–1924.

Whitaker, J. S., and T. M. Hamill, 2012: Evaluating methods to account for system errors in ensemble data assimilation. *Monthly Weather Review*, **140 (9)**, 3078–3089.

Appendix A

Supplementary materials for Chapter 2

A.1 The derivation of the particle flow

In the appendix, we provide the derivation of the particle flow in Equation (2.7). The Kullback-Leibler divergence (KL divergence) between the intermediate pdf at pseudo time s, q_s, and targeted pdf $p(\mathbf{x}|\mathbf{y}_o)$ is given by:

$$KL(q_s) = \int q_s(\mathbf{x}) \log\left(\frac{q_s(\mathbf{x})}{p(\mathbf{x}|\mathbf{y}_o)}\right) d\mathbf{x} \tag{A.1}$$

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

$$\frac{d}{ds}\mathbf{x} = \mathbf{f}_s(\mathbf{x}),\ s \in [0, \infty] \tag{A.2}$$

$$KL(q_{s+\Delta s}) \leq KL(q_s) \tag{A.3}$$

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:

$$\frac{dKL}{ds} \leq 0 \tag{A.4}$$

That is,

$$\begin{aligned}
\frac{dKL}{ds} &= \frac{d}{ds}\int q_s(\mathbf{x}) \log\left(\frac{q_s(\mathbf{x})}{p(\mathbf{x}|\mathbf{y}_o)}\right) d\mathbf{x} \\
&= \int \frac{\partial}{\partial s}\left\{q_s(\mathbf{x}) \log\left(\frac{q_s(\mathbf{x})}{p(\mathbf{x}|\mathbf{y}_o)}\right)\right\} d\mathbf{x} \\
&= \int \frac{\partial q_s(\mathbf{x})}{\partial s}\left\{\log\left(\frac{q_s(\mathbf{x})}{p(\mathbf{x}|\mathbf{y}_o)}\right) + 1\right\} d\mathbf{x} \leq 0
\end{aligned} \tag{A.5}$$

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

$$\frac{\partial q_s(\mathbf{x})}{\partial s} + \nabla_{\mathbf{x}} \cdot (\mathbf{f}_s(\mathbf{x}) q_s(\mathbf{x})) = 0 \tag{A.6}$$

Based on the Liouville equation (Equation A.6), Equation (A.5) can be written as

$$\frac{dKL}{ds} = -\int \nabla_{\mathbf{x}} \cdot (\mathbf{f}_s(\mathbf{x}) q_s(\mathbf{x})) \left\{ \log \left(\frac{q_s(\mathbf{x})}{p(\mathbf{x}|\mathbf{y}_o)} \right) + 1 \right\} d\mathbf{x} \tag{A.7}$$

Integrating by parts, Equation (A.7) can be further written as

$$\frac{dKL}{ds} = \int \mathbf{f}_s(\mathbf{x}) q_s(\mathbf{x}) \cdot \left\{ \log \left(\frac{q_s(\mathbf{x})}{p(\mathbf{x}|\mathbf{y}_o)} \right) + 1 \right\} d\mathbf{x} \tag{A.8}$$

We have used the fact that q_s should approach zero on the boundary in order to assure that the integral of q_s in the whole domain is finite. Equation (A.8) can be further written as

$$\begin{aligned} \frac{dKL}{ds} &= \int \mathbf{f}_s(\mathbf{x}) q_s(\mathbf{x}) \cdot \nabla_{\mathbf{x}} \left\{ \log q_s(\mathbf{x}) - \log p(\mathbf{x}|\mathbf{y}_o) \right\} d\mathbf{x} \\ &= \int \mathbf{f}_s(\mathbf{x}) \cdot \nabla_{\mathbf{x}} q_s(\mathbf{x}) - \mathbf{f}_s(\mathbf{x}) q_s(\mathbf{x}) \cdot \nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y}_o) d\mathbf{x} \\ &= -\int q_s(\mathbf{x}) \nabla_{\mathbf{x}} \cdot \mathbf{f}_s(\mathbf{x}) + \mathbf{f}_s(\mathbf{x}) q_s(\mathbf{x}) \cdot \nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y}_o) d\mathbf{x} \end{aligned} \tag{A.9}$$

where the integration by parts has been used again from line two to line three in Equation (A.9).

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

$$\mathbf{f}_s(\mathbf{x}) = \langle \mathbf{K}(\mathbf{x}, \cdot), \mathbf{f}_s(\cdot) \rangle \in \mathbb{R}^{N_x} \tag{A.10}$$

where $\mathbf{K}$ is the matrix-valued kernel. The bracket $\langle\rangle$ is the inner product in the RKHS. Based on Equation (A.10), the divergence of $\mathbf{f}_s$ can be written as

$$\nabla_{\mathbf{x}} \cdot \mathbf{f}_s(\mathbf{x}) = \nabla_{\mathbf{x}} \cdot \langle \mathbf{K}(\mathbf{x}, \cdot), \mathbf{f}_s(\cdot) \rangle = \langle \nabla_{\mathbf{x}} \cdot \mathbf{K}(\mathbf{x}, \cdot), \mathbf{f}_s(\cdot) \rangle_1 \in \mathbb{R} \tag{A.11}$$

Based on Equations (A.10) and (A.11), the rate of change of the KL-divergence in Equation (A.9) can be written as

$$\begin{aligned} \frac{dKL}{ds} &= -\int q_s(\mathbf{x}) \langle \nabla_{\mathbf{x}} \cdot \mathbf{K}(\mathbf{x}, \cdot), \mathbf{f}_s(\cdot) \rangle_1 + \langle \mathbf{K}(\mathbf{x}, \cdot), \mathbf{f}_s(\cdot) \rangle q_s(\mathbf{x}) \cdot \nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y}_o) d\mathbf{x} \\ &= -\int q_s(\mathbf{x}) \left\{ \langle \nabla_{\mathbf{x}} \cdot \mathbf{K}(\mathbf{x}, \cdot), \mathbf{f}_s(\cdot) \rangle_1 + \langle \mathbf{K}(\mathbf{x}, \cdot) \nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y}_o), \mathbf{f}_s(\cdot) \rangle_1 \right\} d\mathbf{x} \\ &= \langle -\int q_s(\mathbf{x}) \left\{ \nabla_{\mathbf{x}} \cdot \mathbf{K}(\mathbf{x}, \cdot) + \mathbf{K}(\mathbf{x}, \cdot) \nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y}_o) \right\} d\mathbf{x}, \mathbf{f}_s(\cdot) \rangle_1 \end{aligned} \tag{A.12}$$

In order to make the pseudo time rate change of KL-divergence negative, i.e, Equation (A.4), we choose $\mathbf{f}_s$ based on Equation (A.12) as

$$\mathbf{f}_s(\cdot) = \mathbf{D} \int q_s(\mathbf{x}) \left\{ \nabla_{\mathbf{x}} \cdot \mathbf{K}(\mathbf{x}, \cdot) + \mathbf{K}(\mathbf{x}, \cdot) \nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y}_o) \right\} d\mathbf{x} \equiv \mathbf{D}\mathbf{I}_f \tag{A.13}$$

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 (A.12) becomes:

$$\frac{dKL}{ds} = \langle -\mathbf{I}_f, \mathbf{D}\mathbf{I}_f \rangle \leq 0 \tag{A.14}$$

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 (A.13), we need to represent $q_s(\mathbf{x})$ with the particle representation. Assume the particles are $\{\mathbf{x}_s^i\}_{i=1}^{N_p}$, then

$$q_s(\mathbf{x}) = \frac{1}{N_p} \sum_{i=1}^{N_p} \delta(\mathbf{x} - \mathbf{x}_s^i) \tag{A.15}$$

With Equations (A.13) and (A.15), we can obtain the solution of the particle flow:

$$\mathbf{f}_s(\cdot) = \mathbf{D} \sum_{i=1}^{N_p} \nabla_{\mathbf{x}_s^i} \cdot \mathbf{K}(\mathbf{x}_s^i, \cdot) + \mathbf{K}(\mathbf{x}_s^i, \cdot) \nabla_{\mathbf{x}_s^i} \log p(\mathbf{x}_s^i | \mathbf{y}_o) \tag{A.16}$$

where

$$\nabla_{\mathbf{x}_s^i} \cdot \mathbf{K}(\mathbf{x}_s^i, \cdot) = \nabla_{\mathbf{x}} \cdot \mathbf{K}(\mathbf{x}, \cdot)|_{\mathbf{x}=\mathbf{x}_s^i} \tag{A.17}$$

and

$$\nabla_{\mathbf{x}_s^i} \log p(\mathbf{x}_s^i | \mathbf{y}_o) = \nabla_{\mathbf{x}} \log p(\mathbf{x} | \mathbf{y}_o)|_{\mathbf{x}=\mathbf{x}_s^i} \tag{A.18}$$

Appendix B

Supplementary materials for Chapter 3

In this appendix, we describe the details of the actual implementation of the PFF-DART algorithm in this work. Specifically, the details of calculating the particle flow in Equation (2.7), which depends on the choice of kernel, the gradient of the posterior and the preconditioner, will be reported.

B.1 The choice of the kernel

In this study, we choose a kernel as the following

$$\mathbf{K}(\mathbf{x}^i, \mathbf{x}^j) = k(\mathbf{x}^i, \mathbf{x}^j)\mathbf{I}_{N_i \times N_i} \tag{B.1}$$

where k is a scalar kernel and $\mathbf{I}$ is an identity matrix with the size equal to the size of the inner domain N_i. The scalar kernel is chosen as

$$k(\mathbf{x}^i, \mathbf{x}^j) = \exp\left(-\frac{d^{i,j}}{2\alpha}\right) \tag{B.2}$$

where $d^{i,j} = (\mathbf{x}^i - \mathbf{x}^j)^T \mathbf{B}^{-1} (\mathbf{x}^i - \mathbf{x}^j)$ and $\mathbf{B}$ is the prior covariance of the inner domain variables (which is a sub-matrix of the full prior covariance) and α is the kernel width. Although Hu and van Leeuwen (2021) suggests that a matrix-valued kernel can prevent particles from collapsing in a high-dimensional system, using a scalar kernel is good enough in this work because the dimension of the inner domain is typically small. In addition, there are some advantages of using scalar kernel in terms of computational efficiency, which will be detailed later.

Since the size of the inner domain N_i can vary with observations, the kernel width α should also scale with N_i to ensure desirable interactions between particles. Instead of finding an optimal relation between N_i and α, here we propose an adaptive kernel width algorithm to automatically

determine an α in assimilation of each observation. This algorithm depends on only one parameter k_{min}, which sets a minimum kernel value in the first iteration to ensure minimum interaction between the particles that are farthest away. For example, if $k_{min} = 0.1$, we calculate α such that it makes the minimum kernel values between any pair of particles to be 0.1. The way to calculate α is to first calculate all pairs of $d^{i,j}$. The maximum $d_{max} = \max d^{i,j}$ will yield the smallest kernel value. Therefore, α is determined by rearranging Equation (B.2):

$$\alpha = -\frac{d_{max}}{2 \log k_{min}} \tag{B.3}$$

Note that α is determined in the first iteration and will be fixed for the remaining iterations for the assimilation of a single observation (i.e., α is calculated for each scalar observation at each DA cycle, while k_{min} is fixed throughout the whole DA experiment).

We conduct a sensitivity test for PFF-DART with different value of k_{min} in assimilating linear and Gaussian observation (see Section 3-4 for the details of the experiment). In general, Figure B.1 suggests that a larger k_{min}, which corresponds to larger kernel width, will result in a larger ensemble spread. This is because a larger kernel width can lead to a stronger smoothing of the posterior pdf. The result here is very similar to Hu and van Leeuwen (2021).

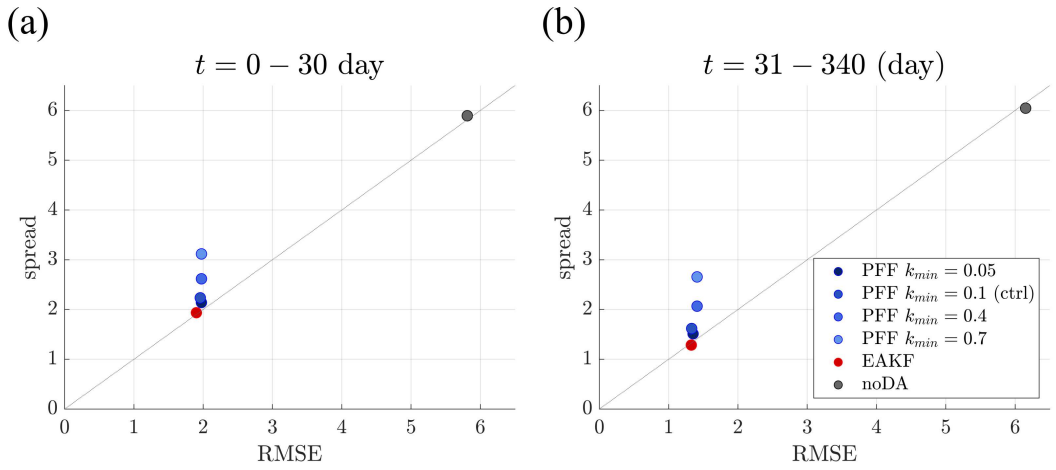

Figure B.1: Similar to Figure 3.4, but for the sensitivity of the PFF-DART to different k_{min}.

B.2 The preconditioner $\mathbf{D}$

Following Hu and van Leeuwen (2021), we choose the prior covariance matrix as the preconditioner. In addition to accelerating the convergence, the choice of $\mathbf{D} = \mathbf{B}$, along with the choice of the scalar kernel defined in Equation (B.2) also has some advantages in terms of computational efficiency. Specifically, we can re-write the particle flow in Equation (2.7) as follows:

$$\mathbf{f}(\mathbf{x}) = \frac{1}{N_p} \sum_{i=1}^{N_p} k(\mathbf{x}, \mathbf{x}^i) \left\{ \mathbf{B} \nabla_{\mathbf{x}^i} \log p(\mathbf{x}^i | \mathbf{y}_o) \right\} + \left\{ \mathbf{B} \nabla_{\mathbf{x}^i} k(\mathbf{x}, \mathbf{x}^i) \right\} \tag{B.4}$$

where $\mathbf{B} \nabla_{\mathbf{x}^i} \log p(\mathbf{x}^i | \mathbf{y}_o)$ can be further simplified as

$$\mathbf{B} \nabla_{\mathbf{x}^i} \log p(\mathbf{x}^i | \mathbf{y}_o) = -(\mathbf{x}^i - \bar{\mathbf{x}}) + \mathbf{B} \nabla_{\mathbf{x}^i} \log p(\mathbf{y}_o | \mathbf{x}^i) \tag{B.5}$$

which reduces the number of matrix vector multiplication since $\mathbf{B}$ and $\mathbf{B}^{-1}$ cancel out. Similarly, $\mathbf{B}^{-1}$ in $\nabla_{\mathbf{x}^i} k(\mathbf{x}, \mathbf{x}^i)$ is cancelled out by the preconditioner, which reduces the number of matrix vector multiplications, i.e., $\mathbf{B} \nabla_{\mathbf{x}^i} k(\mathbf{x}, \mathbf{x}^i)$ can be simplified as

$$\mathbf{B} \nabla_{\mathbf{x}^i} k(\mathbf{x}, \mathbf{x}^i) = -\frac{k(\mathbf{x}, \mathbf{x}^i)}{\alpha} (\mathbf{x}^i - \mathbf{x}^j) \tag{B.6}$$

B.3 The choice of learning rate $\triangle s$

The proper choice of the pseudo-time step, or the learning rate Δs, can be important in determining the efficiency of the PFF. In general, a small increment $\Delta \mathbf{x} = \Delta s \mathbf{f}(\mathbf{x})$ is regarded as a "safe increment" for the PFF iteration. This is because those excessively large steps can make the PFF iterations unstable. On the other hand, a too small, or too safe, increment is not preferred either since it makes PFF convergence slow and inefficient.

Hu and van Leeuwen (2021) used an adaptive scheme to determine Δs, which starts with a fixed but small Δs at the beginning of the iteration, and increases Δs in the later iterations if the magnitude of the particle flow $\|\mathbf{f}(\mathbf{x})\|$ decreases with pseudo-time steps. The idea of this

algorithm is that $\|\mathbf{f}(\mathbf{x})\|$ is usually large at the first several iterations and gradually decreases for later iterations. Therefore, using a small Δs at the beginning when $\|\mathbf{f}(\mathbf{x})\|$ is large ensures that we are not making a giant step. Using a larger Δs at later iterations can accelerate the convergence when $\|\mathbf{f}(\mathbf{x})\|$ becomes small.

One caveat of the adaptive algorithm in Hu and van Leeuwen (2021) is that the optimal Δs at the first iteration can be different for different observations. Certainly, we could choose the smallest first-iteration Δs among all the observations as the "safest Δs" for all the observations. However, this "safest Δs" can then be too small for some observations that do not require such a small Δs, rendering inefficient iterations for these observations. In addition, it can be demanding to find such an *optimal* first-iteration Δs for each type of observation.

Therefore, in this work we propose a revised adaptive learning rate algorithm as follows. The philosophy is to normalize $\|\mathbf{f}(\mathbf{x})\|$ by Δs such that the increment in each iteration has similar size. Denote f_j^i as the j^{th} component ($j = 1, ..., N_i$, where N_i is the size of inner domain) of the particle flow from the i^{th} particle ($i = 1, ..., N_p$). We define a non-dimensional flow by letting each component $\tilde{f}_j^i = \frac{1}{\sigma_{b,j}} f_j^i$, where $\sigma_{b,j}$ is the prior standard deviation of the j^{th} inner domain variable. Then, we calculate the average magnitude of each component of the particle flow over all the particles as

$$\mathcal{N}_j = \sqrt{\frac{1}{N_p} \sum_{i=1}^{N_p} (\tilde{f}_j^i)^2} \tag{B.7}$$

and

$$\mathcal{N} = \sqrt{\frac{1}{N_i} \sum_{j=1}^{N_i} (\mathcal{N}_j)^2} \tag{B.8}$$

where $\mathcal{N}$ is a measure of the average magnitude of the particle flow over all the particles. We choose the learning rate Δs adaptively as follows:

$$\Delta s = \min \left(\frac{\epsilon}{\mathcal{N}}, \Delta s_{max} \right) \tag{B.9}$$

where ϵ is a constant set to 0.05, and Δs_{max} is a parameter that controls the maximum learning rate. By choosing $\Delta s \propto \frac{1}{\mathcal{N}}$, we can make the size of the increment in each iteration similar. The reason we set a upper bound of learning rate as Δs_{max} is that $\mathcal{N}$ can become infinitely small when the PFF almost converges, which can make Δs infintely large, rendering unstable behavior.

We have tested the revised adaptive learning rate algorithm for different types of observations. The same set of parameters work well for all the observations we examined in this study.

Appendix C

Supplementary materials for Chapter 4

C.1 The implicit i.i.d assumption using the ensemble covariance to estimate the background error covariance

The goal of this appendix is to show that, when we use the standard background error subtraction method to estimate the observation error with an ensemble, we also implicitly need the i.i.d assumption (i.e., assume that the ensemble is indistinguishable from the truth). Bonavita et al. (2020) has showed that in an ensemble framework, if we use the ensemble mean $\overline{X}$ as X^b (i.e., $\varepsilon^b \equiv \overline{X} - X^{truth}$) and define the innovation as $D = Y - \overline{H\left(X\right)}$, then we can obtain:

$$D = \left[Y - H\left(X^{truth}\right)\right] + \left[H\left(X^{truth}\right) - \overline{H(X)}\right] = \varepsilon^o + \left[H\left(X^{truth}\right) - \overline{H(X)}\right] \tag{C.1}$$

We can linearize the observation operator to find

$$D \approx \varepsilon^o - \mathbf{H}\left(\overline{X} - X^{truth}\right) = \varepsilon^o - \mathbf{H}\varepsilon^b \tag{C.2}$$

Therefore, we can estimate the observation error covariance based on

$$E\left[(Y - \overline{H(X)})(Y - \overline{H(X)})^T\right] = E\left[\varepsilon^o \varepsilon^{o^T}\right] + E\left[\left(\mathbf{H}\varepsilon^b\right)\left(\mathbf{H}\varepsilon^b\right)^T\right] = \mathbf{R} + \mathbf{HBH}^T \tag{C.3}$$

where

$$\mathbf{B} = E\left[\varepsilon^b \varepsilon^{b^T}\right] \tag{C.4}$$

is the background error covariance. Note that in the ensemble method, we approximate $\mathbf{B}$ with the sample covariance $\mathbf{B}_e$ estimated from the ensemble:

$$\mathbf{B} \approx \mathbf{B}_e = \frac{1}{N_p - 1} \sum_{i=1}^{N_p} \left(x^i - \overline{x}\right) \left(x^i - \overline{x}\right)^T \tag{C.5}$$

where N_p is the ensemble size, x^i is the state vector of the i^{th} member, and $\bar{x}$ is the ensemble mean vector. Comparing Equation (C.4) and (C.5), we can see that

$$E\left[\left(X^{truth} - \overline{X}\right) \left(X^{truth} - \overline{X}\right)^T\right] \approx \frac{1}{N_p - 1} \sum_{i=1}^{N_p} \left(x^i - \overline{x}\right) \left(x^i - \overline{x}\right)^T \tag{C.6}$$

Note that Equation (C.6) implies that, although typically not stated explicitly, we have implicitly assumed that each of the ensemble member x^i is a draw from the random variable X^{truth}. In other words, we do require the i.i.d assumption whenever we use the approximation in Equation (C.5).

C.2 The cost-function and the smoothness parameter in DOEE

In order to find a smooth pdf that can best satisfy Equation (4.19), we introduce prior information in the problem formulation as follows:

$$q\left(\mathbf{f}_{\varepsilon^o}\right) = \left\|\mathbf{A}\mathbf{f}_{\varepsilon^o} - \mathbf{f}_D\right\|^2 + \left(\mathbf{F}\mathbf{f}_{\varepsilon^o}\right)^T \mathbf{S}^{-1} \left(\mathbf{F}\mathbf{f}_{\varepsilon^o}\right) \tag{C.7}$$

$$\min q\left(\mathbf{f}_{\varepsilon^o}\right) \text{ with } \mathbf{f}_{\varepsilon^o} \geq 0 \tag{C.8}$$

where $\mathbf{F}$ is a finite difference operator which calculates the derivative of $\mathbf{f}_{\varepsilon^o}$. Condition (C.8) ensures that all pdf estimates are non-negative, while the constraint with the derivatives (second term on the RHS of Equation (C.7)) enforces a smooth solution. The derivative matrix $\mathbf{F}$ can be

written as:

$$\mathbf{F} = \begin{bmatrix} -1 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & -1 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & -1 & 1 \end{bmatrix} \tag{C.9}$$

which is a one-sided derivative. Note that zero elements of the matrix have been left blank in Equation (C.9). Two sided derivatives, which have rows with elements $(1, -2, 1)$ did not lead to stable solutions. $\mathbf{S}$ is a covariance matrix that controls the smoothness of $\mathbf{F}\mathbf{f}_{\varepsilon^o}$ and is constructed as:

$$\mathbf{S} = \alpha \begin{bmatrix} 1 & r_1 & r_2 & \cdots & \cdots & \cdots & r_{2n} \\ r_1 & 1 & r_1 & r_2 & \cdots & \cdots & r_{2n-1} \\ r_2 & r_1 & 1 & r_1 & r_2 & \cdots & r_{2n-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ r_{2n-1} & \cdots & \cdots & r_2 & r_1 & 1 & r_1 \\ r_{2n} & \cdots & \cdots & \cdots & r_2 & r_1 & 1 \end{bmatrix} \tag{C.10}$$

where α is a scalar controlling the magnitude of $\mathbf{S}$, which is called the "smoothness parameter", and $r_i = \exp\left(-\left(\frac{i\Delta x}{r_{corr}}\right)^2\right)$ in which r_{corr} is chosen as Δx here and it is the correlation length scale for the matrix $\mathbf{S}$.

The first term $\|\mathbf{A}\mathbf{f}_{\varepsilon^o} - \mathbf{f}_D\|^2$ in the cost-function in Equation (C.7) is a measure of the difference between the estimated innovation pdf based on a solution of $\mathbf{f}_{\varepsilon^o}$ and the true innovation pdf, which we would like to minimize. The second term $(\mathbf{F}\mathbf{f}_{\varepsilon^o})^T \mathbf{S}^{-1} (\mathbf{F}\mathbf{f}_{\varepsilon^o})$ is a penalty term, which is analogous to a prior constraint in the cost-function of the variational methods. Therefore, the inverse of $\mathbf{F}^T\mathbf{S}^{-1}\mathbf{F}$ is analogous to the prior covariance matrix, where α determines the prior variance. When α is larger, the constraint from the penalty term is weaker, so we can expect a less smooth $\mathbf{f}_{\varepsilon^o}$, which can, however, fit $\|\mathbf{A}\mathbf{f}_{\varepsilon^o} - \mathbf{f}_D\|^2$ better.

Unfortunately, there is no physically based way to determine the size of the smoothness parameter α. In order to determine an optimal α, we seek the following criteria for a good solution for $\mathbf{f}_{\varepsilon^o}$. First, the solution $\mathbf{f}_{\varepsilon^o}$ should make $\|\mathbf{A}\mathbf{f}_{\varepsilon^o} - \mathbf{f}_D\|$ as small as possible. Second, $\mathbf{f}_{\varepsilon^o}$ is expected to be smooth. Of course, the second criterion is a little ad-hoc and can be problematic when the true underlying distribution is rugged. However, we believe that most of the error distributions that we are looking at are smooth. Based on the two criteria, we examine the relation between the minimum $\|\mathbf{A}\mathbf{f}_{\varepsilon^o} - \mathbf{f}_D\|$ that we can obtain for each value of α, which is shown in Figure C.1. We can see that $\|\mathbf{A}\mathbf{f}_{\varepsilon^o} - \mathbf{f}_D\|$ decreases with increasing α. This is because as α increases, the penalty term becomes smaller (the penalty term vanishes as $\alpha \to \infty$), and minimizing the cost-function becomes almost equivalent to minimizing $\|\mathbf{A}\mathbf{f}_{\varepsilon^o} - \mathbf{f}_D\|$ as $\alpha \to \infty$.Note that $\|\mathbf{A}\mathbf{f}_{\varepsilon^o} - \mathbf{f}_D\|$ never really goes to zero because we also have a constraint $\mathbf{f}_{\varepsilon^o} \geq 0$. Most importantly, Figure C.1 shows that when α is larger than a certain threshold, $\|\mathbf{A}\mathbf{f}_{\varepsilon^o} - \mathbf{f}_D\|$ almost "saturates" and does not decrease much if we further increase α. At the same time, the noise level of $\mathbf{f}_{\varepsilon^o}$ increases if we further increase α. This motivates us to heuristically choose the optimal smoothness parameter $\alpha_{optimal}$ to be around this threshold, since this should be the smoothest $\mathbf{f}_{\varepsilon^o}$ that best fits the equation $\mathbf{f}_D = \mathbf{A}\mathbf{f}_{\varepsilon^o}$, which satisfies both criteria. For example, this threshold should be around 1-100 in Figure C.1.

We demonstrate how the smoothness parameter can affect the solution in an idealized experiment where the true observation error is Gaussian. See Section 4.3 for the details of the experiment setup. Figures C.2 shows the pdf of the actual draws for the observation errors (black dashed line) and the estimated observation errors (blue solid line) using different smoothness parameters α. We can see that when we choose an α that is too small (Fig.C.2a), the estimated distribution is too smooth, whereas a too large α leads to a noisy distribution (Fig.C.2c). When we pick an appropriate smoothness parameter (which is chosen based on the example in Figure C.1), the estimated pdf is a smooth bell-shaped curve that is similar to the true distribution (Fig.C.2b). To understand how α affects the estimation in detail, we plot each of the pdfs in the convolution equation from Equation (4.15) when using a different α in Figure C.3. Note that the innovation pdf (solid red line) and the ensemble difference pdf (orange line) are the input to this methodology and therefore are

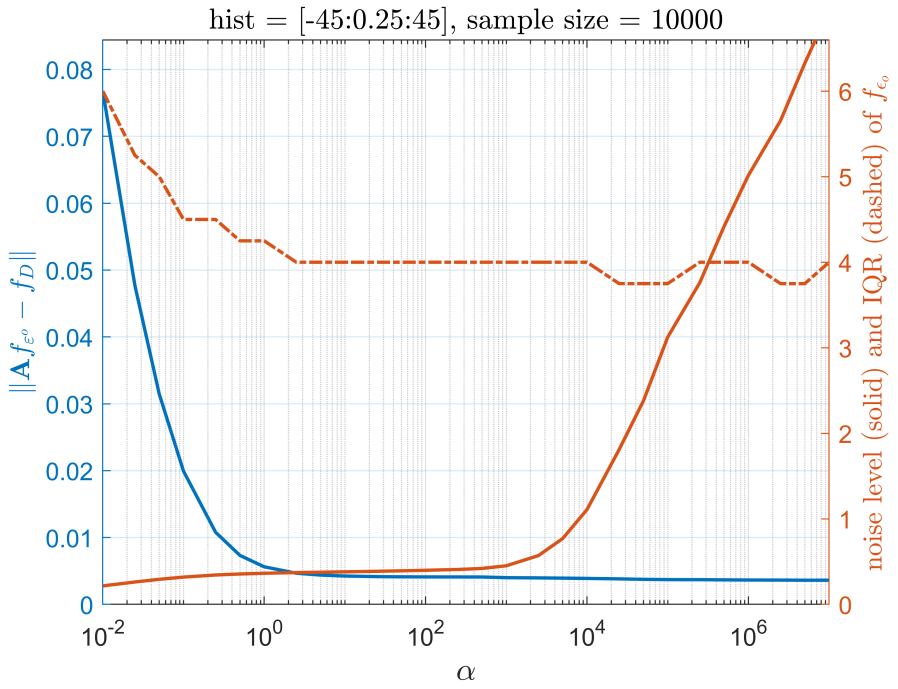

Figure C.1: The statistical properties of the observation error pdf $\mathbf{f}_{\varepsilon^o}$ that minimizes the cost-function $q\left(\mathbf{f}_{\varepsilon^o}\right)$ in Equation (C.7) when $q\left(\mathbf{f}_{\varepsilon^o}\right)$ is constructed by different values of the smoothness parameter α. The blue line is the value of $\|\mathbf{A}\mathbf{f}_{\varepsilon^o} - \mathbf{f}_D\|$, the brown solid line is the noise level of $\mathbf{f}_{\varepsilon^o}$, which is defined as 50 times of the root-mean-square of the derivative of $\mathbf{f}_{\varepsilon^o}$, and the brown dashed line is the interquartile range (IQR) of $\mathbf{f}_{\varepsilon^o}$.

the same for all the subplots in Figure C.3. We also plot the reconstructed innovation pdf (dashed brown line) obtained from the convolution of the estimated observation error pdf (blue line) and the ensemble difference pdf (orange line). With a small α (Fig.C.3a), the difference between the reconstructed innovation pdf and the actual innovation pdf and is large, which corresponds to the large value of $\|\mathbf{A}\mathbf{f}_{\varepsilon^o} - \mathbf{f}_D\|$ in Figure C.1. In contrast, when α is larger than the $\alpha_{optimal}$, the reconstructed innovation pdf becomes very similar to the actual innovation pdf (Figs.C.3b-c), which corresponds to the small values of $\|\mathbf{A}\mathbf{f}_{\varepsilon^o} - \mathbf{f}_D\|$ in Figure C.1. Note that although both of the reconstructed innovation pdfs in Figures C.3b,c fit the actual innovation pdf very well, the observation error pdf in Figure C.3c is rugged, which seemingly manages to fit the noise in the actual observation error (black dashed line in Figure C.2). The results here justify that the way we choose $\alpha_{optimal}$ tends to give us good estimates, as long as the true observation error pdf is smooth.

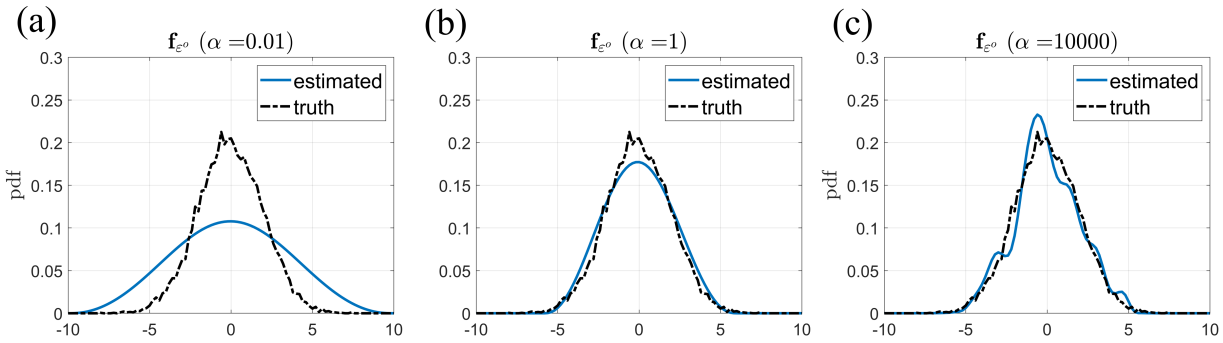

Figure C.2: The true observation error pdf (black dashed line), which is a Gaussian $Normal\left(0, 2^2\right)$, and the estimated (blue solid line) observation error pdf when the estimate is based on (a) a too small smoothness parameter α (b) an appropriate smoothness parameter and (c) a too large α.

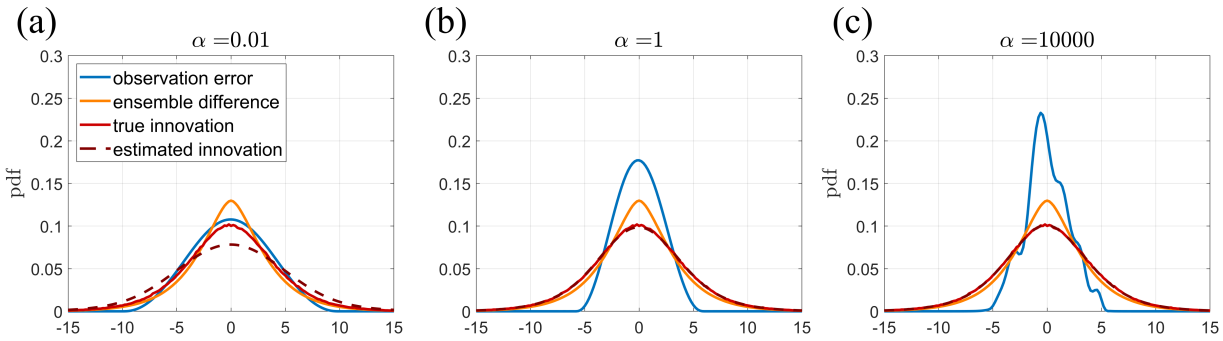

Figure C.3: The decomposition of the pdfs in Equation (4.15) when the true observation error is drawn from $Normal\left(0, 2^2\right)$. The blue line is the estimated observation error pdf f_{ε^o} (the same as the blue lines in Fig.C.2), the orange line is the ensemble difference pdf $f_{H\left(X^{truth}\right)-H(X)}$, the solid red line is the innovation pdf f_D, and the dashed brown line is the estimated innovation pdf given by the convolution between the estimated observation error pdf f_{ε^o} and the ensemble difference pdf $f_{H\left(X^{truth}\right)-H(X)}$. The estimate is obtained based on (a) a too small smoothness parameter α (b) an appropriate smoothness parameter and (c) a too large α.

Appendix D

Supplementary materials for Chapter 5

D.1 Making the cost-function quadratic

In this appendix, we provide more mathematical details of how we linearize the cost-function at around $\mathbf{x}_0$ in Equation (5.2). With Gaussian prior and Gaussian observation error, the cost-function $J(\mathbf{x})$ can be written as

$$J(\mathbf{x}) = \frac{1}{2}\left\{(\mathbf{x}-\mathbf{x}_b)^T\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}_b) + (\mathbf{y}-H(\mathbf{x}))^T\mathbf{R}^{-1}(\mathbf{y}-H(\mathbf{x}))\right\} \tag{D.1}$$

Let $J(\mathbf{x}) = \frac{1}{2}\mathbf{f}(\mathbf{x})^T\mathbf{f}(\mathbf{x}) = \frac{1}{2}\|\mathbf{f}(\mathbf{x})\|^2$, where

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} \mathbf{B}^{-1/2}(\mathbf{x}-\mathbf{x}_b) \\ \mathbf{R}^{-1/2}(\mathbf{y}-H(\mathbf{x})) \end{pmatrix} \tag{D.2}$$

and define

$$Q_0(\Delta\mathbf{x}) \equiv \frac{1}{2}\|\mathbf{f}(\mathbf{x}_0) + \nabla\mathbf{f}(\mathbf{x}_0)\Delta\mathbf{x}\|^2 \tag{D.3}$$

where Q_0 is the linearized quadratic cost-function that we minimize in each outer loop.

To better understand specifically where the linearization is done, we can define $\mathbf{x} = \mathbf{x}_0 + \Delta\mathbf{x}$, and re-write the cost-function as a function of $\Delta\mathbf{x}$:

$$\begin{aligned} J(\Delta\mathbf{x}) = \frac{1}{2}\{&(\mathbf{x}_0-\mathbf{x}_b+\Delta\mathbf{x})^T\mathbf{B}^{-1}(\mathbf{x}_0-\mathbf{x}_b+\Delta\mathbf{x}) \\ &+ (\mathbf{y}-H(\mathbf{x}_0+\Delta\mathbf{x}))^T\mathbf{R}^{-1}(\mathbf{y}-H(\mathbf{x}_0+\Delta\mathbf{x}))\} \end{aligned} \tag{D.4}$$

Note that when H is linear, $J(\Delta\mathbf{x})$ is already quadratic and there is no need to linearize the cost-function. When H is non-linear, we can linearize $H(\mathbf{x}_0+\Delta\mathbf{x})$ as

$$H(\mathbf{x}_0 + \Delta\mathbf{x}) \approx H(\mathbf{x}_0) + \mathbf{H}^T \Delta\mathbf{x} \tag{D.5}$$

where $\mathbf{H}^T = \nabla H(\mathbf{x}_0)$ is the adjoint of the observation operator evaluated at $\mathbf{x}_0$. Plug Equation (D.5) into Equation (D.4), i.e.,

$$\begin{aligned} J(\Delta\mathbf{x}) &= \frac{1}{2}\left\{(\widetilde{\mathbf{x}}_b + \Delta\mathbf{x})^T\mathbf{B}^{-1}(\widetilde{\mathbf{x}}_b + \Delta\mathbf{x}) + (\widetilde{d} - \mathbf{H}^T\Delta\mathbf{x})^T\mathbf{R}^{-1}(\widetilde{d} - \mathbf{H}^T\Delta\mathbf{x}))\right\} \\ &\equiv Q_0(\Delta\mathbf{x}) \end{aligned} \tag{D.6}$$

where $\widetilde{\mathbf{x}}_b = \mathbf{x}_0 - \mathbf{x}_b$ and $\widetilde{d} = \mathbf{y} - H(\mathbf{x}_0)$, and we observe that Equation (D.6) becomes quadratic. Therefore, for Gaussian prior and Gaussian observation errors, the linearization of the cost-function is mainly in linearizing the observation operator.

With some algebra, it can be shown that Equation (D.6) is equivalent to Equation (D.3).

D.2 The conditions for the convergence to the mode of posterior using the evolving-Gaussian method

In this appendix, we try to prove that for a Gaussian prior and linear H, the evolving Gaussian method will converge to the mode of the posterior pdf under certain conditions. In this proof, we assume the background in the observation space y_b is larger than the mode of the likelihood y_o (Note that y_o is the observation if the mode of the observation error is zero). In this case, the analysis will satisfy $y_o < y_a < y_b$. Note that the proof is analogous for the other way around where $y_o > y_a > y_b$.

As mentioned in Section 5.3.4, the criteria for the evolving-Gaussian method to converge to y_a for monotonic $g(y)$ are as follows:

1. When $g(y)$ increases from y_o to y_b (Figure D.1a), it is required that there exists only one intersection between $h(y)$ and $g(y)$.

2. When $g(y)$ decreases from y_o to y_b (Figure D.1b), it is required that the transformation $F = (f \circ f)$ must satisfy

$$if\ y > y_a,\ then\ F(y) < y$$

and

$$if\ y < y_a,\ then\ F(y) > y$$

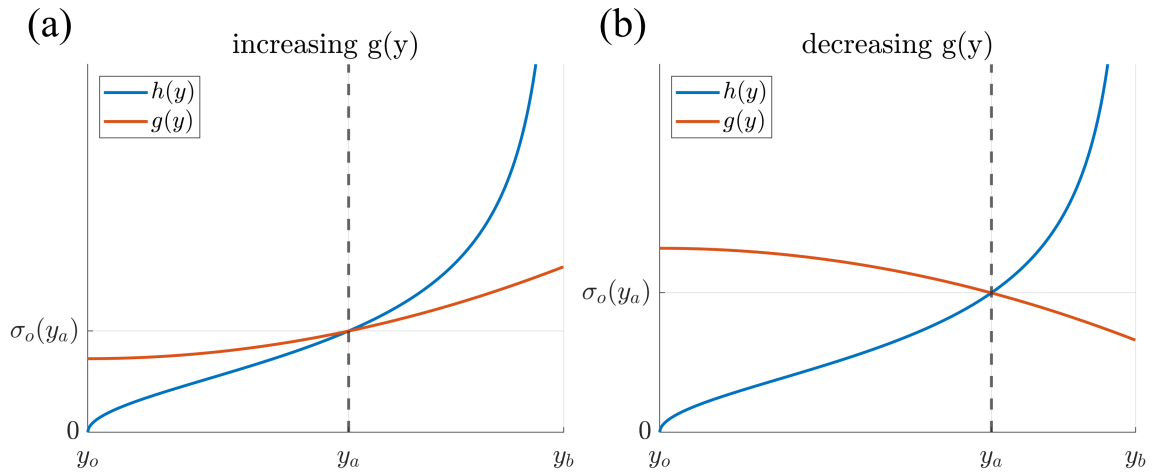

Figure D.1: The relation between the steady curve $h(y)$ and monotonic (a) increasing and (b) decreasing $g(y)$.

Recall that when $g(y)$ decreases from y_o to y_b, there must exist only one intersection between $h(y)$ and $g(y)$. In other words, for those cases where the evolving-Gaussian method converges, $h(y)$ and $g(y)$ have only one intersection. At the intersection, $g(y) = h(y)$, and therefore $f(y) = y$ (from Equation (5.17)), i.e., the evolving Gaussian method will stop when it reaches the intersection between $h(y)$ and $g(y)$. In this case, the intersection will be the analysis y_a. This is because for the original non-Gaussian *uni-modal* cost-function, the mode is the *only* point where the gradient vanishes. In the evolving Gaussian method, $f(y) = y$ suggests that the point y is the point with zero gradient. Since the evolving Gaussian method uses *exactly the same gradient* as the original cost-function, the point such that $f(y) = y$ is the mode of the original non-Gaussian cost-function

y_a. Therefore, we can also write the intersection, or the analysis y_a as

$$y_a = \left(\frac{g(y_a)^2}{g(y_a)^2 + \sigma_b^2}\right) y_b + \left(\frac{\sigma_b^2}{g(y_a)^2 + \sigma_b^2}\right) y_o \tag{D.7}$$

and note that $h(y_a) = g(y_a)$ and $f(y_a) = y_a$.

In the following analysis, it is more convenient to observe the behavior of the errors $\{r^n\}$:

$$r^n = y^n - y_a \tag{D.8}$$

Based on Equation (5.14) and (D.7), r^n can be re-arranged as:

$$r^n = \frac{\sigma_b^2(y_b - y_o)(g(y^{n-1})^2 - g(y_a)^2)}{(g(y^{n-1})^2 + \sigma_b^2)(g(y_a)^2 + \sigma_b^2)} \tag{D.9}$$

and note that $\{r^n\}$ converges if and only if $\{y^n\}$ converges.

D.2.1 When $g(y)$ increases from y_o towards y_b

An example of such $g(y)$ is shown in Figure D.1(a). We first claim that $\{r^n\}$ is monotonically decreasing and bounded below, i.e., $\{r^n\}$ converges, and therefore $\{y^n\}$ converges. Then, we claim that $\{y^n\}$ converges to y_a.

- Claim: $\{r^n\} \geq 0$ and is monotonically decreasing.

  First observe that $r^0 = y^0 - y_a = y_b - y_a > 0$. Note that based on Equation (D.9)

  $$r^n \sim (g(y^{n-1})^2 - g(y_a)^2) \tag{D.10}$$

  where $\sim$ means the LHS and the RHS has the same sign. Since $g > 0$ and is monotonically increasing,

  $$(g(y^{n-1})^2 - g(y_a)^2) \sim (y^{n-1} - y_a) \tag{D.11}$$

Therefore, we can observe that

$$r^0 > 0$$
$$r^1 \sim (y^0 - y_a) = r^0 > 0$$
$$r^2 \sim (y^1 - y_a) = r^1 > 0$$

It is easy to show that $r^n > 0$ for all n by induction. To see $\{r^n\}$ is decreasing, consider:

$$r^{n+1} - r^n = \frac{\sigma_b^2(y_b - y_o)(g(y^n)^2 - g(y^{n-1})^2)}{(g(y^n)^2 + \sigma_b^2)(g(y^{n-1})^2 + \sigma_b^2)} \sim (g(y^n)^2 - g(y^{n-1})^2) \tag{D.12}$$

and we can observe that

$$r^1 - r^0 = y^1 - y_b < 0$$
$$r^2 - r^1 \sim (g(y^1)^2 - g(y^0)^2) \sim y^1 - y^0 = r^1 - r^0 < 0$$
$$r^3 - r^2 \sim (g(y^2)^2 - g(y^1)^2) \sim y^2 - y^1 = r^2 - r^1 < 0$$

By induction we can easily see that $r^{n+1} - r^n < 0$ for all n. Therefore, we have proved that $\{r^n\}$ is decreasing and bounded below ($\because r^n > 0$), so we conclude that $\{r^n\}$ converges, or equivalently, $\{y^n\}$ converges.

- Claim: $\{y^n\} \to y_a$

  We use the method of proof by contradiction here. Suppose $\{y^n\} \to \tilde{y}_a \neq y_a$. Note that it is impossible for $\tilde{y}_a < y_a$ because we have already proved that $r^n > 0$, or equivalently $y^n > y_a$. Therefore, we assume $\tilde{y}_a > y_a$. By the definition of the convergence, given an arbitrary $\varepsilon > 0$, there exists an integer N such that as $n \geq N$, we have

$$y^n < \tilde{y}_a + \varepsilon \tag{D.13}$$

  Here we let $\varepsilon = f^{-1}(\tilde{y}_a) - \tilde{y}_a$, where $f^{-1} = (g^{-1} \circ h)$ from Equation (5.17). Note that since $\tilde{y}_a > y_a$, $h(\tilde{y}_a) > g(\tilde{y}_a)$ (this is the result from the assumption that there is only

one intersection between h and g, see Figure D.2). Since g is monotonically increasing, $f^{-1}(\tilde{y}_a) = g^{-1}(h(\tilde{y}_a)) > \tilde{y}_a$, and therefore $\varepsilon > 0$.

Before proceeding the proof, we first observe that f is monotonically increasing because for $x \leq y$, $g(x) \leq g(y)$, and therefore $f(x) \leq f(y)$. If we apply f to Equation (D.13), we obtain

$$y^{n+1} = f(y^n) < f(\tilde{y}_a + \varepsilon) = \tilde{y}_a \tag{D.14}$$

which is contradictory to the assumption that $\{y^n\} \to \tilde{y}_a$ (because $\{y^n\}$ is monotonically decreasing and we find y^{n+1} is smaller than $\tilde{y}_a$, so $\tilde{y}_a$ cannot be the lower bound). We exclude the possibility that $\{y^n\}$ converge to a number $\tilde{y}_a > y_a$. Therefore, $\{y^n\}$ must converge to y_a.

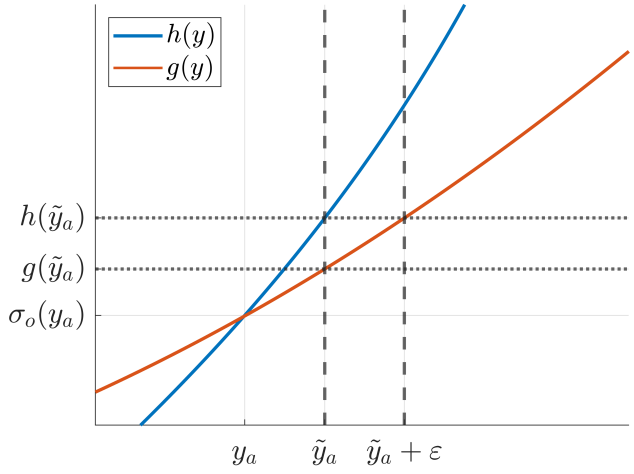

Figure D.2: The demonstration of how ε is chosen given an arbitrary $\tilde{y}_a > y_a$.

D.2.2 When $g(y)$ decreases from y_o towards y_b

An example of such $g(y)$ is shown in Figure D.1(b). The strategy is to prove that the two sub-sequences: $\{y^{2n}\}$ and $\{y^{2n+1}\}$ both converge to y_a, using the similar ideas as in the previous subsection.

- Claim: $\{r^{2n}\} \geq 0$ and is monotonically decreasing; $\{r^{2n+1}\} \leq 0$ and is monotonically increasing.

  Similarly, based on Equation (D.10), we can observe that

$$
\begin{aligned}
&r^0 > 0 \\
&r^1 \sim (g(y^0)^2 - g(y_a)^2) \sim -(y^0 - y_a) = -r^0 < 0 \\
&r^2 \sim (g(y^1)^2 - g(y_a)^2) \sim -(y^1 - y_a) = -r^1 > 0 \\
&r^3 \sim (g(y^2)^2 - g(y_a)^2) \sim -(y^2 - y_a) = -r^2 < 0
\end{aligned}
$$

  By induction, we can see $r^{2n} > 0$ and $r^{2n+1} < 0$. Similar to Equation (D.12), we can derive:

$$
r^{n+2} - r^n \sim (g(y^{n+1})^2 - g(y^{n-1})^2) \tag{D.15}
$$

  Similarly, observe that

$$
\begin{aligned}
&r^2 - r^0 < 0 \\
&r^3 - r^1 \sim (g(y^2)^2 - g(y^0)^2) \sim -(y^2 - y^0) = -(r^2 - r^0) > 0 \\
&r^4 - r^2 \sim (g(y^3)^2 - g(y^1)^2) \sim -(y^3 - y^1) = -(r^3 - r^1) < 0 \\
&r^5 - r^3 \sim (g(y^4)^2 - g(y^2)^2) \sim -(y^4 - y^2) = -(r^4 - r^2) > 0
\end{aligned}
$$

  Consequently, by induction, we can conclude that $\{r^{2n}\}$ is decreasing and $\{r^{2n+1}\}$ is increasing.

  In summary, for $\{r^{2n}\}$, it is monotonically decreasing and bounded below, thus converges. For $\{r^{2n+1}\}$, it is monotonically increasing and bounded above, and hence converges as well.

- Claim: $\{y^{2n}\} \to y_a$ and $\{y^{2n+1}\} \to y_a$ for certain conditions

  We note that $\{y^{2n}\}$ and $\{y^{2n+1}\}$ do not always converge to y_a. We add an additional condition to ensure this here. For convenience, we define $F = (f \circ f)$, such that $F(y^n) =$

$f(f(y^n)) = f(y^{n+1}) = y^{n+2}$. The additional condition to ensure convergence to y_a is as follows

$$if\ y > y_a,\ then\ F(y) < y \tag{D.16}$$

and

$$if\ y < y_a,\ then\ F(y) > y \tag{D.17}$$

Note that we have already proved that $y^n > y_a$ and $y^{n+2} = F(y^n) < y^n$ when n is even. Similarly, $y^n < y_a$ and $y^{n+2} = F(y^n) > y^n$ when n is odd. However, we note that the above condition is stronger: we require *all* $y > y_a$ to satisfy $F(y) < y$ and *all* $y < y_a$ to satisfy $F(y) > y$.

We start the proof with the even sub-sequence. Assume $\{y^{2n}\} \to \tilde{y}_a \neq y_a$. Since $\{y^{2n}\} > y_a$ so it is impossible for $\tilde{y}_a < y_a$. Therefore, assume $\tilde{y}_a > y_a$. Let $\varepsilon = F^{-1}(\tilde{y}_a) - \tilde{y}_a$. Similar to previous subsection, we can verify that F is monotonically increasing. Note that based on the criterion (D.16), $F(\tilde{y}_a) < \tilde{y}_a$ and $\tilde{y}_a < F^{-1}(\tilde{y}_a)$ so $\varepsilon > 0$. Therefore, by definition, for this ε we can find N such that as $2n \geq N$, we have

$$y^{2n} < \tilde{y}_a + \varepsilon$$

Similarly, we can apply F to the above inequality to obtain

$$y^{2n+2} = F(y^{2n}) < F(\tilde{y}_a + \varepsilon) = \tilde{y}_a$$

Therefore, we conclude that $\{y^{2n}\}$ cannot converge to $\tilde{y}_a > y_a$. Therefore, $\{y^{2n}\} \to y_a$.

For the odd sub-sequence, assume $\{y^{2n+1}\} \to \tilde{y}_a \neq y_a$. Since $\{y^{2n+1}\} < y_a$ so it is impossible for $\tilde{y}_a > y_a$. Therefore, assume $\tilde{y}_a < y_a$. Let $\varepsilon = \tilde{y}_a - F^{-1}(\tilde{y}_a)$. Note that based on the criterion (D.17), $F(\tilde{y}_a) > \tilde{y}_a$, and $\tilde{y}_a > F^{-1}(\tilde{y}_a)$ so $\varepsilon > 0$. Therefore, by

definition, for this ε we can find N such that as $2n+1 \geq N$, we have

$$y^{2n+1} > \tilde{y}_a - \varepsilon$$

Similarly, we can apply F to the above inequality to obtain

$$y^{2n+3} = F(y^{2n+1}) > F(\tilde{y}_a - \varepsilon) = \tilde{y}_a$$

Therefore, we conclude that $\{y^{2n+1}\}$ cannot converge to $\tilde{y}_a < y_a$. Therefore, $\{y^{2n+1}\} \to y_a$.

D.3 A pratical way to estimate the standard deviation

Due to the limited sample size and the choice of bin-size for the histogram, the estimated pdfs from real data can sometimes be noisy, making the estimated standard deviation σ_o from Equation (5.11) noisy as well. To obtain a smoother pdf, we perform a regression to fit local points with a quadratic function to get a smoothed standard deviation. Specifically, we use a quadratic function centered at the mode of the pdf, and find the (two remaining) coefficients to fit the local points (here we collect points for ± 3 K) in a least square manner. Note that only the coefficient of the square term is related to the slope. For example, if we would like to estimate the standard deviation at $d = 5K$, and the histogram from the data shows the mode is at $d = 1K$. Then, we try to find coefficient p_2 and p_0 from the quadratic function $p_2(x-1)^2 + p_0$ to fit the points collected between $d = 2K$ to $d = 8K$. The estimated standard deviation is then calculated based on $\sigma_o = \sqrt{\frac{1}{2p_2}}$.

In order to simplify the representation of these non-Gaussian errors in IFS, we reduce the parameters that needs to be recorded by fitting a piecewise linear function to the estimated σ_o from the quadratic regression. Specifically, we first calculate the average σ_o around the mode, and then fit two linear regression lines to the left and to the right of the mode respectively (the black line in Figure D.3a,d). Note that the estimated σ_o at the tails can be spurious due to the small number of samples, therefore when performing the linear regressions we remove the values in the

10 rightmost and leftmost bins with non-zero pdf values. Therefore, for each non-Gaussian pdf, we only record four parameters in this work:

1. the mode of the pdf
2. the average standard deviation σ_o around the mode
3. the slope of σ_o right of the mode
4. the slope of σ_o left of the mode

Figure D.3 demonstrates two examples of the non-Gaussian pdfs estimated from the horizontally polarized 19 GHz radiances. For the first case where the original pdf is a Gaussian-like but with a wider tail (Figure D.3a-c), we see the piecewise linear fit for σ_o is a good representation of the direct estimate from quadratic regression, except for the tails (Figure D.3a). We reconstruct the cost-function and the pdf from the piecewise linear σ_o and compare to the original ones in Figure D.3b,c. The reconstructed curves resemble the original curves in the first example. The second example shows a skewed and multi-modal pdf. Since we only deal with unimodal pdf in this work, we choose the mode as the one with the largest pdf value. In Figure D.3d, we see the estimated σ_o goes to infinity in the other two modes around $d = -25$ and $d = 20$. Nevertheless, the linear regression ignores the other two modes and tries to capture the overall features of the pdf. Figure D.3e,f show the reconstructed pdfs from the piecewise linear fit smooth out the two other modes. In particular, the gradient of the cost-function from the piecewise linear fit is the average of the gradient of the original cost-function around the other two modes.

Finally, recall that we are constructing the pdfs for the error model depending on $\overline{C_{37}}$. The above procedure stores the four parameters for a single non-Gaussian pdf. Since we separtely estimate the pdf for $\overline{C_{37}}$ in ten bins, we have these four parameters for these ten bins. Although unnecessary, we perform an additional cubic fit to the four parameters as a function of $\overline{C_{37}}$ in the final step to obtain a smooth transition of the parameters between different cloud bin. Note that the method described here is just one practical but somewhat crude way to represent the non-Gaussian error model, and we can expect a possibly improved method in the future. Nevertheless,

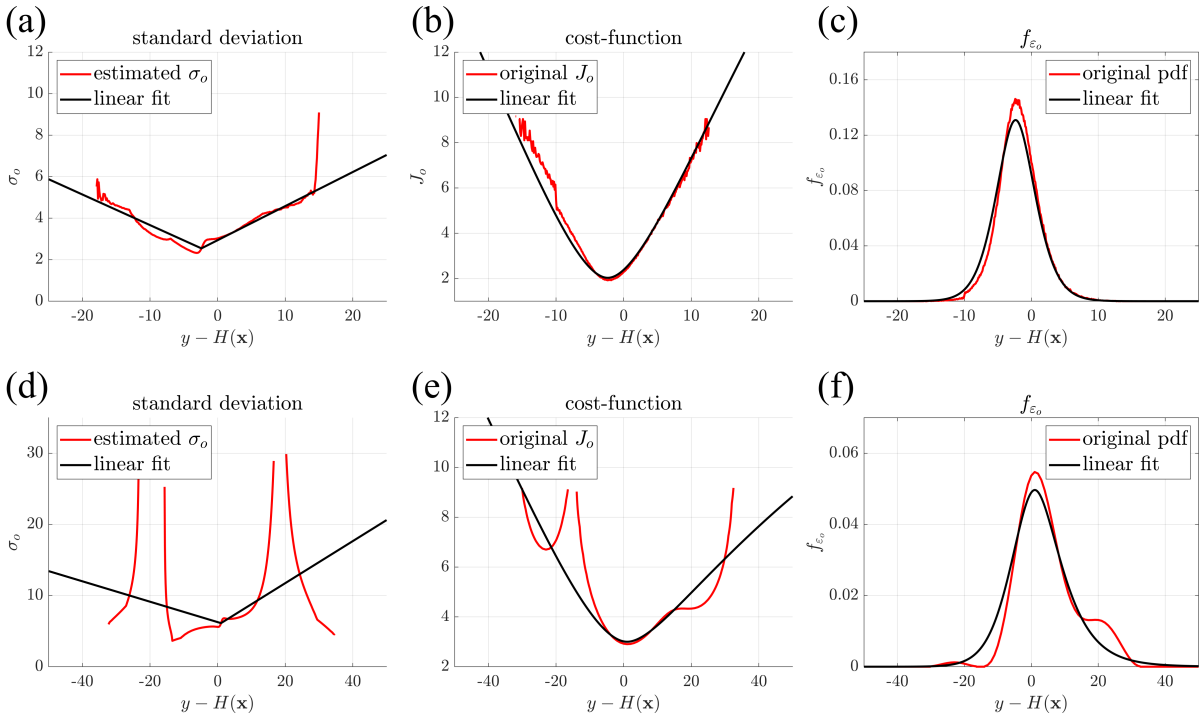

Figure D.3: The demonstration of (a)(d) the estimated standard deviation $\sigma_o(d)$, (b)(e) the cost-function $J_o(d) = -\log f_{\varepsilon^o}(d)$, and (c)(f) non-Gaussian error pdf $f_{\varepsilon^o}(d)$, as a function of innovation $d = y - H(\mathbf{x})$. The red curves are the original pdfs and the estimation from the quadratic regression, and the black curves are constructed from the piecewise linear fit. (a)-(c) is the innovation pdf of the 19H radiance when $\overline{C_{37}} \in [0.0, 0.1]$ and (d)-(f) is the observation error pdf estimated from DOEE of the 19H radiance when $\overline{C_{37}} \in [0.2, 0.3]$

we emphasize that there are practical advantages to store as few number of parameters as possible. Also, given the reconstructed pdfs and cost-functions, we consider the way we estimate σ_o to represent the non-Gaussian errors can capture the overall shape and skewness of the pdfs.