A PDE-based Adaptive Kernel Method for Solving Optimal Filtering Problems

Zezhong Zhang Richard Archibald[y] Feng Bao[z]

Abstract

In this paper, we introduce an adaptive kernel method for solving the optimal ltering problem. The computational framework that we adopt is the Bayesian lter, in which we recursively generate an optimal estimate for the state of a target stochastic dynamical system based on partial noisy observational data. The mathematical model that we use to formulate the propagation of the state dynamics is the Fokker-Planck equation, and we introduce an oper-ator decomposition method to eciently solve the Fokker-Planck equation. An adaptive kernel method is introduced to adaptively construct Gaussian kernels to approximate the probability distribution of the target state. Bayesian inference is applied to incorporate the observational data into the state model simulation. Numerical experiments have been carried out to validate the performance of our kernel method.

Keywords: Optimal ltering problem, Bayesian inference, partial dierential equation, kernel approximation

1 Introduction

Data assimilation is an important topic in data science. It aims to optimally combine a mathematical model with observational data. The key mission in data assimilation is the optimal ltering problem, in which we try to nd the best estimate for the state of a stochastic dynamic model. Such a stochastic dynamic model is typically in the form of a system of stochastic dierential equations (SDEs), and we call it the \state process". In many practical situations, the true value of the state process is not available, and we can only use partial noisy observations to nd the best estimate for the state. In the optimal ltering problem, the \best estimate" that we want to nd is dened as the conditional expectation of the state conditioning on the observations.

When both the state dynamics and the observations are linear, the optimal ltering problem is a linear ltering problem, which can be analytically solved by the classic Kalman lter [20]. In the case of nonlinear ltering problem, one needs to derive an approximation for the conditional probability of the state { instead of calculating the conditional expectation directly, and we call this conditional probability the \ltering density". There are two well-known nonlinear ltering methods, e.g., the Zakai's approach and the Bayesian lter. The Zakai's approach formulates the ltering density as the solution of a parabolic type stochastic partial dierential equation [32], e.g., the Zakai equation, then we solve the Zakai equation numerically to obtain an approximation for the ltering density [5, 6, 14, 34]. The Bayesian lter solves the nonlinear ltering problem in a recursive

Department of Mathematics, Florida State University, Tallahassee, Florida.

[y]Computational Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee, (archibaldrk@ornl.gov).

[z]Department of Mathematics, Florida State University, Tallahassee, Florida, (bao@math.fsu.edu).

two-step procedure: a prediction step and an update step. In the prediction step, we predict the ltering density of the state before reception of the observational data. Then, when the observational data is available in the update step, we let the predicted ltering density be the prior and apply Bayesian inference to incorporate the observational information into the prediction and obtain an updated ltering density as the posterior in the Bayesian inference. Besides the Zakai's approach and the Bayesian lter, recently we developed a new approach to solve the optimal ltering problem, and we named this approach the backward SDE lter [4, 9]. The backward SDE lter is similar to Zakai's approach in the sense that it utilizes a system of dierential equations (e.g., the backward SDEs) to analytically propagate the ltering density [3, 8]. At the same time, taking advantage of the SDE nature, the backward SDE lter is highly scalable, which can be implemented eciently on parallel computers.

Among the aforementioned approaches, the Bayesian lter is widely used for solving the nonlinear ltering problem in practice. The state-of-the-art Bayesian lter methods include the particle lter [1,16] and the ensemble Kalman lter [12,13]. Both the particle lter and the ensemble Kalman lter use samples (particles) to create an empirical distribution to describe the predicted ltering density (i.e. the prior) in the prediction step. In the update step, the particle lter applies Bayesian inference to assign weights to particles and use weighted particles to represent the updated ltering density (i.e. the posterior). On the other hand, the ensemble Kalman lter linearizes the observations and then adopts the Kalman update in the Kalman lter method to derive an updated ltering density. Since the Kalman lter is designed for linear problems, in the case that the optimal ltering problem is highly nonlinear, the ensemble Kalman lter does not provide accurate estimates for the target state [29, 30]. The major drawback of the particle lter is the degeneracy issue [21, 27]. When the observational data lies on the tail of the predicted ltering density, only very few particles will receive high likelihood weights in the Bayesian inference procedure, which signicantly reduces the eective particle size in the particle lter.

In addition to the methodology of using particles to propagate the state process, another approach that can transport the probability density forward in time is to solve the Fokker-Planck equation, which is a parabolic type particle dierential equation (PDE). Although the PDE-based Fokker-Planck approach analytically formulates the propagation of the state model, solving a PDE in high dimensional state space is computationally expensive, which makes PDE-based optimal ltering solvers dicult to be practically applied.

In this work, we will develop a novel kernel method to eciently solve the Fokker-Planck equation, and the kernel approximated solution for the Fokker-Planck equation will be used as our estimate for the predicted ltering density in the prediction step. Then, we will adopt Bayesian inference to incorporate the observational data into the kernel approximated ltering density.

Kernel method recently attracted extensive attentions in machine learning and function approximation [15,18]. When solving the optimal ltering problem, the target function that we approximate with kernels is the ltering density, which is a probability density function (PDF). In many scenarios in the optimal ltering problem, the ltering density appears to be a bell-shaped function. This makes kernel method (especially with Gaussian type kernels) an eective way to construct approx-imations for the target ltering density [2]. Since optimal ltering is often used to solve practical application problems in real time, eciency of an optimal ltering method is essential. In this pa-per, we will introduce an operator decomposition method to decompose PDF propagation in the Fokker-Planck equation into a linear component and a nonlinear component. The linear component of propagation can be analytically derived, and the nonlinear component needs to be carried out numerically. Numerical solver for the nonlinear component in the Fokker-Planck equation will be formulated as an optimization problem, which aims to determine kernel parameters that describe the nonlinear propagation of the ltering density. To implement the optimization procedure e-ciently, we will introduce a boosting algorithm [23] to adaptively generate kernels to capture the main features of the state distribution. This allows us to use minimum amount of active kernels to characterize the ltering density, which will be used to estimate the target state.

The rest of this paper is organized as follows. In Section 2, we introduce some preliminaries that we need to design the PDE-based adaptive kernel method for solving the optimal ltering problem. Then, we shall give a detailed description for our adaptive kernel method in Section 3. Numerical examples that validate the eectiveness of the kernel approach in solving the optimal ltering problem and comparison experiments will be presented in Section 4. Finally, summary and concluding remarks will be given in Section 5.

2 Preliminaries

In this section, we provide the preliminaries to formulate our adaptive kernel method for solving the optimal ltering problem. We shall rst briey introduce the optimal ltering problem. Then, we will discuss one of the most important optimal ltering approaches, i.e., the Bayesian lter, and we will describe the mathematical framework of the adaptive kernel method as a Bayesian lter type approach.

2.1 The optimal ltering problem

In the optimal ltering problem, we consider the following stochastic dynamical system in the form of a stochastic dierential equation (SDE) in the probability space (
; F ; P)

$$dX_t = b(t; X_t)dt + {}_t dW_t; \qquad (1)$$

where $b : R^+ \; R^d \; ! \; R^d$ is the drift coecient, $: R^+ \; ! \; R^{dr}$ is the diusion coecient of the SDE, W is a standard r-dimensional Brownian motion under P, and the ${}_t dW_t$ term is a standard Itô type stochastic integral, which brings additive noises to the dynamical model. The d-dimensional stochastic process $X := fX_t g_{t0}$ is called the \state process", which represents the state of the dynamical model. In order to estimate the state of X_t when the true value of X_t is not available, we collect partial noisy observational data for X_t, denoted by Y_t, which is dened by

$$Y_t = h(X_t)dt + dB_t; \qquad (2)$$

where $h : R^d \; ! \; R^l$ is an observation function that measures the state of X_t and B is another Brownian motion independent of W with covariance R at any given time t. The stochastic process Y is often called the \observation process".

The goal of the optimal ltering problem is to nd the best estimate for (X_t) given the observational information Y_t, where $Y_t := (Y_s; 0 \; s \; t)$ is the -algebra generated by the observation process Y , and is a given test function. In mathematics, the best estimate for (X_t) is dened by the \optimal lter", denoted by $\tilde{}(X_t)$, which is the conditional expectation of (X_t), i.e.

$$\tilde{}(X_t) := E[\;(X_t)jY_t]: \qquad (3)$$

In this paper, we focus on the case that f (in the state process) and/or h (in the observation process) are nonlinear functions. The linear ltering problem is well-solved by the Kalman lter (except for the extremely high dimensional cases). To solve the nonlinear optimal ltering problem, the standard approach aims to estimate the conditional probability of the state, i.e. $P(X_t jY_t)$, which is also called the \ltering density". Then, we can calculate the conditional expectation in Eq. (3) through the integration formula

$$E[\;(X_t)jY_t] = \int \;(x)P(xjY_t)dx:$$

In what follows, we will introduce the Bayesian lter, which provides a two-step procedure to estimate the ltering density $P(X_t jY_t)$ recursively.

2.2 The recursive Bayesian lter

The Bayesian lter recursively estimates the target state X_t on a sequence of discrete time instants $0 = t_1 < t_2 < \; ; < t_n < \;$, and the Bayesian lter framework is composed of two steps: the prediction step and the update step.

Prediction step.

Assume that the ltering density $p(X_{t_n}|Y_{t_n})$ is given at time t_n. In the prediction step, we propagate the ltering density from time t_n to time t_{n+1} without usage of the new observational data $Y_{t_{n+1}}$, and we want to get the predicted ltering density, i.e. $p(X_{t_{n+1}}|Y_{t_n})$.

There are three major methods to achieve this goal:

The rst method is designed to nd the predicted ltering density through the following Chapman-Kolmogorov formula

$$p(X_{t_{n+1}}|Y_{t_n}) = \int p(X_{t_{n+1}}|X_{t_n})p(X_{t_n}|Y_{t_n})dX_{t_n};$$

where $p(X_{t_{n+1}}|X_{t_n})$ is the transition probability of the state equation (1) that transports the previous ltering density $p(X_{t_n}|Y_{t_n})$ from t_n to t_{n+1}. The above Chapman-Kolmogorov formula is often carried out by independent sample simulations, and it's the primary prediction technique in particle-based optimal ltering methods, such like the particle lter and the ensemble Kalman lter. As a result of the particle propagation of the ltering density, one may obtain an empirical representation for the predicted ltering density.

The second method utilizes the following (time-inverse) backward stochastic dierential equation (BSDE) to generate the predicted ltering density:

$$P_{t_{n+1}} = P_{t_n} \quad \int_{t_n}^{t_{n+1}} \sum_{i=1}^{d} \frac{\partial b_i}{\partial x_i}(X_t)P_t dt \quad \int_{t_n}^{t_{n+1}} Q_t dW_t; \qquad P_{t_n} = p(X_{t_n}|Y_{t_n});$$

where X_t is the state process, and the $\int_{t_n}^{t_{n+1}} dW_t$ is a backward Itô integral, which is an Itô type stochastic integral integrated backwards [7, 24]. The solutions of the above BSDE is a pair $(P; Q)$, where Q is the martingale representation of P with respect to W [11]. We refer to [4, 8, 9] for more details of the BSDE method.

The third method, which is also the method that we are going to discuss in this paper, describes the propagation of the ltering density through the following Fokker-Planck equation over the time interval $[t_n; t_{n+1}]$

$$\frac{\partial p(x;t)}{\partial t} = \sum_{i=1}^{d} \frac{\partial}{\partial x_i} b_i(x;t)p(x;t) + \sum_{i;j=1}^{d} \frac{\partial^2}{\partial x_i \partial x_j} D_{i;j}\; p(x;t) \tag{4}$$

with initial condition $p(x;t_n) = p(X_{t_n} = x|Y_{t_n})$, where b_i is the i-th component of the drift function b, and the matrix D is dened by $D = \frac{1}{2}\;>$. As a result, solution $p(x;t_{n+1})$ of the Fokker-Planck equation (4) gives us the desired predicted ltering density $p(X_{t_{n+1}} = x|Y_{t_n})$.

Update step.

With an approximation for the predicted ltering density (obtained through either one of the aforementioned method), the Bayesian lter updates the predicted ltering density to the (posterior) ltering density via the following Bayesian inference formula

$$p(X_{t_{n+1}}|Y_{t_{n+1}}) = \frac{p(X_{t_{n+1}}|Y_{t_n})\; p(Y_{t_{n+1}}|X_{t_{n+1}})}{p(Y_{t_{n+1}}|Y_{t_n})}; \tag{5}$$

where

$$p(Y_{t_{n+1}}|X_{t_{n+1}}) = \exp\left(-\frac{\left(Y_{t_{n+1}} - h(X_{t_{n+1}})\right)^2}{R}\right) \tag{6}$$

is the likelihood function, and $p(Y_{t_{n+1}}|Y_{t_n})$ in the denominator normalizes the ltering density at the time instant t_{n+1}.

Then, we carry out the above prediction-update procedure recursively to propagate the ltering density $p(X_t|Y_t)$ over time.

3 Adaptive Kernel Approximation Approach

In this paper, we solve the Fokker-Planck equation (4) numerically to generate an approximation for the predicted ltering density $p(X_{t_{n+1}}|Y_{t_n})$, and we apply the Bayesian inference (5) to calculate the estimated (posterior) ltering density $p(X_{t_{n+1}}|Y_{t_{n+1}})$. In what follows, we shall give detailed discussions on the computational framework that we construct to apply the adaptive kernel approximation method to solve the optimal ltering problem.

3.1 Prediction through Fokker-Planck equation

For convenience of presentation, we denote

$$\mathcal{L}_{b,\sigma}p_t := -\sum_{i=1}^{d}\frac{\partial}{\partial x_i}\left(b_i(x,t)p(x,t)\right) + \sum_{i,j=1}^{d}\frac{\partial^2}{\partial x_i \partial x_j}\left(D_{i,j}\, p(x,t)\right),$$

and we call $\mathcal{L}_{b,\sigma}$ the the Fokker-Planck operator in this paper. The Prediction step in our adaptive kernel approximation approach will focus on deriving a numerical solver for the Fokker-Planck equation

$$\frac{\partial p(x,t)}{\partial t} = \mathcal{L}_{b,\sigma}p, \tag{7}$$

and the numerical solution to Eq. (7) will be our approximation to the predicted ltering density, which will be combined with the likelihood function to generate an estimated posterior ltering density.

Numerical methods for solving parabolic type PDEs, such like the Fokker-Planck equation, have been extensively studied [10, 17, 22, 31]. However, when the dimension of the problem is high, solving Eq. (7) becomes an extremely expensive computational task [33]. The primary challenge in obtaining numerical solutions to the Fokker-Planck equation is how to eciently and eectively implement spatial dimensional approximation. Traditional mesh-based numerical methods, such like nite dierence methods and nite element methods typically utilize polynomial approximations to describe solutions of the equation. However, due to the so-called \curse of dimensionality", the computational cost of solving the Fokker-Planck equation increases exponentially as the dimension of the problem increases.

In this work, we adopt the following kernel approximation scheme to approximate the solution of the Fokker-Planck equation

$$p(x,t_n) \approx \sum_{k=1}^{K}\psi_n^k(x), \tag{8}$$

where K is the total number of kernels, and

$$\psi_n^k(x) := \omega_n^k \exp\left(-\frac{1}{2}(x-\mu_n^k)^\top(\Sigma_n^k)^{-1}(x-\mu_n^k)\right)\ {}_n \tag{9}$$

is a Gaussian type kernel function, which is parameterized by weight $|_n^k$, mean $_n^k$ and covariance matrix ^k_n Numerical analysis results have been derived to verify that the kernel approximation scheme (8) is capable of generating accurate approximations to wide range of function when the number of kernels K is suciently large [19, 26, 28]. The reason that we pick Gaussian type kernels to approximate the target function p is that p is the ltering density, which describes a conditional probability distribution. In many situations in the optimal ltering problem, the ltering density is a bell-shaped function, which can be eectively approximated by Gaussian type functions.

Then, assuming that we have a kernel approximation p_n for the ltering density $p(X_{t_n} jY_{t_n})$, we introduce the following temporal discretization scheme to solve the Fokker-Planck equation

$$\tilde{p}_{n+1} = p_n + L_{b;} p_n \ t_n; \quad (10)$$

where $t_n := t_{n+1} \quad t_n$ is the time step-size, and $\tilde{p}_{n+1}$ is a kernel approximation for the predicted ltering density $p(X_{t_{n+1}} \ Y_{t_n})$. Given kernels $f^k_n g^K_{k=1}$ for the approximated ltering density $\tilde{p}_n$ and the approximation scheme

$$p_n := \sum_{k=1}^{K} {}^k_n(x); \quad (11)$$

we can rewrite Eq. (10) as

$$\tilde{p}_{n+1} = \sum_{k=1}^{K} {}^k_n(x) + t_n \ L_{b;} \quad \sum_{k=1}^{K} {}_n^k(x) \ ; \quad (12)$$

and we let

$$\tilde{p}_{n+1} := \sum_{k=1}^{K} \ {}^{\sim k}_{n+1};$$

where $f^{k\sim}_{n+1} g^K_{k=1}$ is a set of kernels that approximates $\tilde{p}_{n+1}$. We can see from the temporal discretization scheme (12) that obtaining an approximation $\tilde{p}_{n+1}$ for the predicted ltering density $p(X_{t_{n+1}} \ Y_{t_n})$ is equivalent to nding parameters for kernels $f^k \sim_{n+1} g^K_{k=1}$. Note that the kernels $f^k_n g^K_{k=1}$ on the right hand side of Eq. (12) are Gaussian (as introduced in Eq. (9)). Hence the Fokker-Planck operator part, i.e. $L_{b;} \quad \sum_{k=1}^{K} {}_n^k(x)$ can be derived analytically. In this way, we transfer the computational cost of solving the Fokker-Planck equation from high dimensional spatial approximation to solving an optimization problem for kernel parameters.

Since the target function for kernel approximation is a PDF, a relatively small number of Gaussian kernels may be sucient to provide a reasonable description for the ltering density. On the other hand, solving the Fokker-Planck equation through Eq. (12) suers from the stability issue. When values of the drift function b in the state equation Eq. (1) are large (or the time step-size t_n is large), the drift term will generate a strong force that pushes the ltering density far from its current location. However, due to exponential decay of Gaussian tails, which would typically cause local behaviors of Gaussian kernels, the ltering density approximated by the kernel approximation scheme (11) can only be transported to a limited distance. This can make our method dicult to track targets driven by state equations with large drift terms.

In the following subsection, we shall introduce an operator decomposition method to alleviate the above stability issue.

3.2 Operator decomposition

The central idea of our operator decomposition method is to divide the Fokker-Planck operator into a drift operator and a diusion operator. Then, we further decompose the drift operator into a linear component and a nonlinear component, and we provide analytical and numerical methods to characterize the linear component and the nonlinear component separately.

Before we introduce our decomposition strategy, we would like to point out the following facts of the Fokker-Planck operator:

Fact 1. Given a PDF p, in the case that the diusion coecient does not contain state X, we have

$$L_{b;}p = L_{b;0}p + L_{0;}p:$$

Fact 2. The Fokker-Planck operator $L_{b;}$ is linear, i.e., for two constants a, b and two PDFs p, q, we have

$$L_{b;}[ap + bq] = aL_{b;}p + bL_{b;}q:$$

Therefore, the kernel approximated ltering density under the Fokker-Planck operator can be written as

$$L_{b;}p_n = \sum_{k=1}^{K} L_{b;}{}_n(x);{}^k$$

and the right hand side of Eq. (12) becomes

$$\sum_{k=1}^{L} {}^k_n(x) + t_n \sum_{k=1}^{K} L_{b;}{}^k_n(x) = \sum_{k=1}^{K} {}_n^k(x) + t_n\; L_{b;}{}^k(x) \;:_n \qquad (13)$$

The linearity of the Fokker-Planck operator allows us to discuss the propagation of each Gaussian kernel separately.

In light of Fact 1, we can handle the drift term rst and then incorporate diusion into the state propagation. Fact 2 allows us to discuss state propagation kernel-by-kernel when necessary.

In this work, instead of deriving the operator decomposition method directly under the numerical PDE framework, we rst switch back to the state equation, and we consider the following Euler-Maruyama scheme that propagates each kernel ${}_n^k$ through the state equation

$$X^k_{n+1} = X^k_n + b(t_n; X^k_n)t_n + {}_{t_n} W_{t_n}; \qquad k = 1; 2; \; ; K; \qquad (14)$$

where the initial state $X^k_n \;\; {}^k_n$, i.e. X^k_n follows the distribution of the k-th Gaussian kernel, and $W_{t_n} := W_{t_{n+1}} \;\; W_{t_n} \;\; N(0; t_n \; I_d)$. In this way, by combining distributions for $fX^k_{n+1}g^K_{k=1}$ obtained through the discretized SDE scheme (14), we get a description for the predicted ltering density, which can also be considered as an approximation for the right hand side of Eq. (13).

To address the stability issue through operator decomposition and to transport Gaussian kernels eectively to the next time step, we introduce a linear approximation to the (nonlinear) drift function, and we denote it by $b^L(t_n; X^k_n) := AX^k_n + \;$, where $A \; 2 \; R^{dd}$ and $\; 2 \; R^d$. The linear operator b^L will be determined as the best linear approximation to b in the sense of least square. In other words, we aim to nd A and that will minimize the mean square error between the original drift function b and the linear approximation b^L, i.e.

$$\min_{A;} E \quad b(t_n; X^k_n) \quad (AX^k_n + \;) \quad {}^2 \; : \qquad (15)$$

To maintain the nonlinearity of the state dynamics, we introduce a residual function $b^N(t_n; X^k_n) := b(t_n; X^k_n) \quad b^L(t_n; X^k_n)$ that models the nonlinear component of b. Hence, the drift function is decomposed into a linear component b^L and a nonlinear component b^N, i.e., $b(t_n; X^k_n) = b^L(t_n; X^k_n) +$

$b^N(t_n; X_n^k)$, and the Euler-Maruyama scheme for the state equation can be interpreted as

$$X_{n+1}^k = X_n^k + \left(b^L(t_n; X_n^k) + b^N(t_n; X_n^k) \right) t_n + W_{t_n}:$$

In what follows, we will introduce a three-step operator decomposition procedure to compute the predicted ltering density $\tilde{p}_{n+1}$.

In the rst step, we only transport the ltering density via the linear component b^L. Specically, we implement the following scheme

$$\begin{aligned} X_{n+1}^{k;L} &:= X_n^k + b^L(t_n; X_n^k) t_n \\ &= (A t_n + I) X_n^k + {}^n t_n \end{aligned} \tag{16}$$

to propagate the ltering density at the time step t_n, and we let

$$T(X_n^k) := (A t_n + I) X_n^k + t_n$$

be the operator that formulates the linear component of the drift function, i.e. $X_{n+1}^{k;L} = T(X_n^k)$. Note that a linear function will map a Gaussian distribution to a Gaussian distribution. Since X_n^k follows a Gaussian distribution, $X_{n+1}^{k;L}$ will also follow a Gaussian distribution, which can be determined by the linear operator $T()$, and we denote the distribution for $X_{n+1}^{k;L}$ by $p_{n+1}^{k;L}$.

In the second step, we incorporate the nonlinear component b^N of the drift function to the ltering density so that both the linear and the nonlinear components are considered in the ltering density propagation. Since b^N does not linearly propagates $X_{n+1}^{k;L}$, we can not derive a Gaussian kernel directly from $p_{n+1}^{k;L}$ to obtain a kernel that describes the nonlinear component of the drift. In order to derive a kernel approximation for the predicted ltering density, which have considered the nonlinear component of the drift, we dene an operator

$$b^{N;T}(t_n; X_{n+1}^{k;L}) := b^N(t_n; T^{-1}(X_{n+1}^{k;L})) = b^N(t_n; X_n^k):$$

Then, with Gaussian distributions $\{p_{n+1}^{k;L}\}_{k=1}^K$ that describe random variables $\{X_{n+1}^{k;L}\}_{k=1}^K$ (introduced in Eq. (16)), we introduce the following PDE type solver to calculate a distribution $\hat{p}_{n+1}$ dened by

$$\hat{p}_{n+1} = \sum_{k=1}^{K} \left(p_{n+1}^{k;L} + \mathcal{L}_{b^{N;T};0}\, p_{n+1}^{k;L} t_n \right); \tag{17}$$

where $\mathcal{L}_{b^{N;T};0}$ is a Fokker-Planck operator with drift $b^{N;T}$, and the diusion is chosen as 0. The PDF $\hat{p}_{n+1}$ on the left hand side of Eq. (17) is an approximation for the predicted ltering density before incorporation of the diusion term, and we use kernel approximation scheme to represent $\hat{p}_{n+1}$, i.e.

$$\hat{p}_{n+1} = \sum_{k=1}^{K} \hat{}_{n+1}^{k}(x); \tag{18}$$

where $\{\hat{}_{n+1}^{k}\}_{k=1}^{K}$ is a set of Gaussian kernels, and we will introduce the procedure to determine parameters for $\{\hat{}_{n+1}^{k}\}_{k=1}^{K}$ in the next subsection.

Finally, in the third step we add diusion back to the predicted ltering density. Since we assume that the state dynamics are perturbed by additive noises in this work, for each Gaussian kernel $\hat{}_{n+1}^{k}$ that approximates $\hat{p}_{n+1}$ in Eq. (18), we can simply introduce the extra diusion information by adding $t_n \, {}_{t_n} {}_{t_n}^{>}$ to the covariance of $\hat{}_{n+1}^{k}$ and get a kernel $\tilde{}_{n+1}^{k}$ to approximate the predicted ltering density at time stage t_{n+1}. As a result, we obtain the kernel approximation for the predicted ltering density as follows

$$\tilde{p}_{n+1} = \sum_{k=1}^{K} \tilde{\ }^{k}_{n+1} : \tag{19}$$

In the above three-step procedure, we can see that the rst step and the third step can be implemented analytically, and the second step incorporates the nonlinear behavior of the dynamical model, which needs an optimization procedure to determine kernel parameters. In what follows, we will introduce an adaptive boosting algorithm to achieve this goal.

3.3 Adaptive boosting algorithm for kernel training

Recall that each kernel in Eq. (18) is Gaussian and has the expression

$$\hat{\ }^{k}_{n+1}(x) = \hat{\ }^{k}_{n+1} \exp\left(-\frac{1}{2}(x - \hat{\ }^{k}_{n+1})^{\top}(\hat{\ }^{k}_{n+1})^{-1}(x - \hat{\ }^{k}_{n+1})\right) :$$

Our optimization procedure aims to nd kernel parameters $\{(\hat{\ }^{k}_{n+1}; \hat{\ }^{k}_{n+1}; \hat{\ }^{k}_{n+1})\}_{k=1}^{K}$ so that the left hand side of Eq. (17), which is determined by $\{\hat{\ }^{k}_{n+1}\}_{k=1}^{K}$ will be equal to the right hand side, which is dened by the linear transformed Gaussian distributions $\{p^{k;L}_{n+1}\}_{k=1}^{L}$. We denote

$$g_{n+1} := \sum_{k=1}^{K} p^{k;L}_{n+1} + L_{b_{N;T};0}\, p^{k;L}_{n+1}\, t_n \tag{20}$$

for convenience of presentation. Since $\{p^{k;L}_{n+1}\}_{k=1}^{K}$ are Gaussian functions, g_{n+1} dened in Eq. (20) can be derived analytically.

In this work, instead of nding all the kernel parameters at the same time by solving a large scale optimization problem, we adopt the so-called \boosting algorithm", which sequentially minimizes the approximation error. Specically, we introduce the Boosting Algorithm in Table 1 to determine the parameter set $\{(\hat{\ }^{k}_{n+1}; \hat{\ }^{k}_{n+1}; \hat{\ }^{k}_{n+1})\}_{k=1}^{K}$.

The boosting algorithm introduced in Table 1 will adaptively generate kernels, and this adaptive kernel approximation procedure allows us to capture more important features (modes) in the ltering density. Also, the Gaussian tails of the kernels can provide reasonable description for low density regions in the ltering density, which will make our method stable.

3.4 Bayesian update for ltering density

To incorporate the observational information to the predicted ltering density, we apply Bayesian inference (5). Since the predicted ltering density is described by multiple kernels, we apply Bayesian inference to each Gaussian kernel and obtain a kernel for the posterior ltering density. Specically, for each state point x, let

$$p^{k;post}_{n+1}(x) = \tilde{\ }^{k}_{n+1}(x)\, p(Y_{t_{n+1}} x);$$

where $p(Y_{t_{n+1}} x)$ is the likelihood function introduced in Eq. (6) with a given state position $X_{t_{n+1}} = x$ and $\tilde{\ }$ is a Gaussian kernel in Eq. (19) that approximates the predicted ltering density $\tilde{p}_{n+1}$. In this way, the entire posterior ltering density is approximated by

$$p^{post}_{n+1} = \sum_{k=1}^{K} p^{k;post}_{n+1} : \tag{21}$$

Table 1: Boosting Algorithm

Algorithm 1: Boosting algorithm to adaptively generate kernels.

```
Initialize the kernel approximation as $\hat{p}_{n+1}(x) = 0$; dene target function $g_{n+1}$ through Eq. (20); set global approximation tolerance tol.
while $k = 1; 2; \; ; K$, do
    - Generate $M$ global state samples, denoted by $f\hat{x}^{(m)}_{n+1} g^{M}_{m=1}$, from the kernel approximated distribution based on $f p^{k;L}_{n+1} g^{K}_{k=1}$.
    - Evaluate the approximation error on each state sample and calculate $e_m := g(\hat{x}_m) \; \hat{p}_{n+1}(\hat{x}_m)$ for $m = 1; 2; \; ; M$.
    - Compute global error $E_g = \frac{1}{M} \sum_{m=1}^{M} (e_m)^2$. If $E_g <$ tol, break and set weights for other kernels 0, i.e. $!_j = 0, k < j \; K$. Otherwise, continue.
    - Locate the state sample with the largest approximation error, i.e. nd $m$ s.t. $e_{\bar{m}} = \max_m e_m$.
    - Generate a Gaussian kernel $\hat{}^{k}_{n+1}$ centered at the state sample that suers from the largest error, i.e. choose the initial guess for the mean as $\hat{}^{k}_{n+1} = \hat{x}_{\bar{m}}$.
    - Solve a local optimization problem to determine the weight and covariance for the kernel $\hat{}^{k}_{n+1}$ by comparing values of $\hat{}^{k}_{n+1}$ (treated as the left hand side of Eq. (17)) with $g_{n+1}$ on locally generated state samples near the kernel center $\hat{}^{k}_{n+1}$.
    - Add the locally trained kernel $\hat{}^{k}_{n+1}$ to kernel approximation $\hat{p}_{n+1}$, i.e. let $\hat{p}_{n+1} = \hat{p}_{n+1} + \hat{}^{k}_{n+1}$.
end while
```

Note that each kernel $p^{k;post}_{n+1}$ that we use to approximate the overall posterior ltering density p^{post}_{n+1} may not be Gaussian due to the nonlinear observation. To derive an approximation by Gaussian kernels, we train a new set of Gaussian kernels to describe the posterior ltering density. Specically, we introduce a kernel approximation

$$p_{n+1} := \sum_{k=1}^{K} {}^{k}_{n+1};$$

and we let p_{n+1} be an approximation to the approximated posterior ltering density p^{post}_{n+1}, i.e. $p_{n+1} \; p^{post}_{n+1}$. To this end, we adopt the same Boosting Algorithm framework introduced in Table 1 again to adaptively generate Gaussian kernels $f^{k}_{n+1} g^{K}_{k=1}$, and a normalization procedure will be implemented to $f^{k}_{n+1} g^{K}_{k=1}$ to make p_{n+1} a PDF.

3.5 Summary of the algorithm

In this subsection, we summarize our algorithm in Table 2.

Table 2: Summary of the algorithm

```
Algorithm 2: Algorithm of the adaptive kernel method.

  Initialize the ltering density $p_0$ with kernels $f^k g_{k=1}^K$.
For $n = 0; 1; 2; 3;$
    Prediction Step:
        - Generate $f p_{n+1}^{k;L} g_{k=1}^K$ through Eq. (16) to incorporate the linear component
            (determined through Eq. (15) ) of the drift function.
        - Use the Boosting Algorithm described in Table 1 to incorporate the nonlinear
            component of the drift function and generate Gaussian kernels $f^{k\wedge}{}_{n+1} g_{k=1}^K$
            from $g_{n+1}$ (dened in Eq. (20) ) to approximate $\hat{p}_{n+1}$.
        - Add $t_{n t_n}{}^{>}$ to$_{t_n}$ the covariance of each Gaussian kernel $_{n+1}\hat{}^k$ to incorporate state
            diusion and get the kernel $^k \tilde{}_{n+1}$ to approximate the predicted ltering
            density $\tilde{p}_{n+1}$ via Eq. (19).
    Update Step
        - Carry out Bayesian inference to generate a posterior ltering density $p_{n+1}^{post}$ de-
            ned in Eq. (21).
        - Carry out Boosting Algorithm in Table 1 again to obtain a Gaussian kernel
            approximation $p_{n+1} = f_{n+1}^k g_{k=1}^K$ to approxiomate $p_{n+1}^{post}$
        - Normalize $f^k{}_{n+1} g_{k=1}^K$ to make $p_{n+1}$ a PDF, and $p_{n+1}$ is the estimated ltering
            density at time stage $n + 1$.
end
```

4 Numerical Experiments

In this section, we present three numerical examples to demonstrate the performance of our adaptive kernel method for solving the optimal ltering problem. We rst present a demonstration example to show how our adaptive kernel approximation method will adaptively capture the main features of the ltering density in state propagation. In the second example, we solve a benchmark optimal ltering problem, i.e. the bearing-only tracking problem, and we compare our method with the particle lter method [25] and the ensemble Kalman lter method [13] to show accuracy and eciency of the adaptive kernel method. Then, in Example 3 we solve a high dimensional Lorenz-96 tracking problem, which is a well-known challenging optimal ltering problem due to the chaotic behavior of the state model.

4.1 Example 1: Demonstration for adaptive kernel approximation.

We use the rst numerical example to demonstrate the performance of our adaptive kernel approximation method in propagating state dynamics. Instead of solving an entire optimal ltering problem, we only present the eectiveness of our method in transporting a probability distribution through the Fokker-Planck equation, and the primary computational eort of our approach lies on using kernels to approximate the Fokker-Planck operator. Since the ltering density is approximated by Gaussian kernels, Gaussian type diusions can be directly added to the target distribution. Therefore, in this example we shall focus on the drift part of the Fokker-Planck operator, i.e. $L_{b;0}$, and the drift term is dened by the following 2D function:

$$b(x_1; x_2) = \quad x_0^{2\,2} \; + \quad 3 \; 4 \quad x_2^{1} \; + \quad 3_2 \; :$$

For convenience of presentation, we consider state propagation in time with step-size be 1. Then, we choose the initial distribution as a standard Gaussian distribution, denoted by , and we apply the drift operator $L_{b;0}$ to . In this way, the target function that we try to use our kernel method to approximate is $F \; := \; + L_{b;0}$. In Figure 1, we present the original target function F driven by the operator $L_{b;0}$ on left, and the linear approximation for function F obtained by the linear transportation Eq. (16) is presented on the right. From this gure, we can see that the linear component can roughly capture the main feature of the target function.

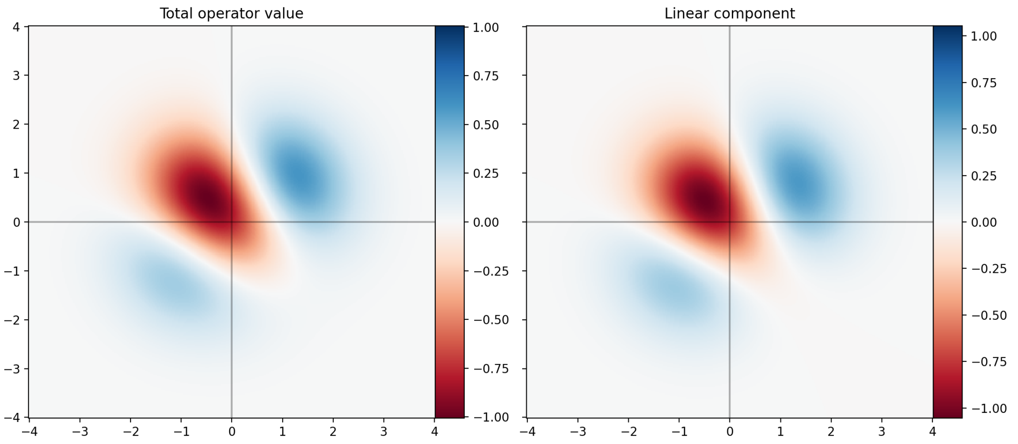

Figure 1: Example 1. Linear component in describing the Fokker-Planck operator

To demonstrate the performance of kernel method in approximating the nonlinear component (described in Eq. (17)) of the operator and the eectiveness of the adaptive boosting algorithm, we compare the analytically derived true nonlinear component of the function with the approximated nonlinear component in Figure 2. The subplot on the left shows the true function that we aim to approximate, and the subplot on the right is our approximated function by using the boosting algorithm introduced in Table 1. We use blue-to-red colors to represent function values, and we can see from this gure that the boosting algorithm can accurately capture the true function, which describes the nonlinear component of the Fokker-Planck operator.

To show more details of the performance of the adaptive kernel construction in the boosting algorithm, we present the approximation errors after tting up to 6 kernels in Figure 3. From this gure, we can see that by using only one kernel to describe the nonlinear component of the Fokker-Planck operator, the main part of the function in the region $[\;\; 1; 0] \;\; [\;\; 1; 1]$ (presented in the left subplot in Figure 2) is well tted, and two remaining features that represent two tails in the function (plotted in Figure 2) need to be tted. Then, by adding the second and the third kernels, we can successfully approximate those two tails and get low overall tting errors. As more and more kernels are added, we get rid of higher error regions one-by-one. As a result, we obtain more and more accurate approximations to the nonlinear component of the drift operator.

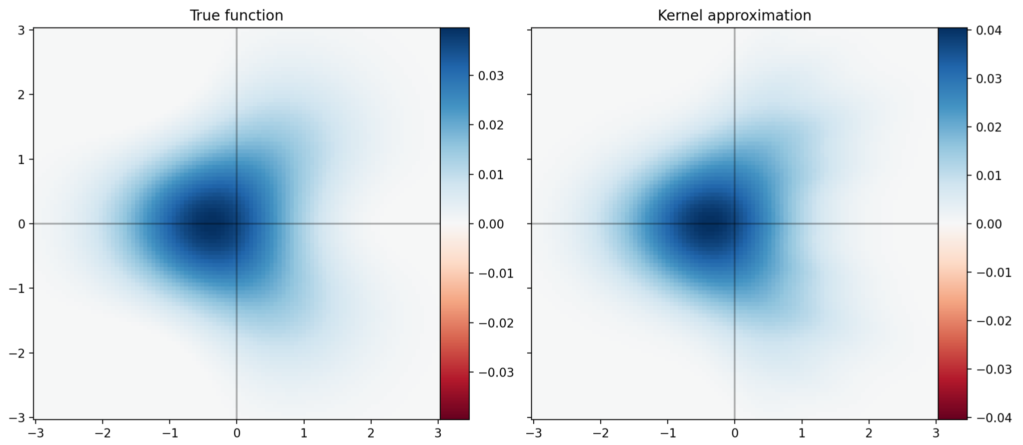

Figure 2: Example 1. Accuracy of approximation obtained by the boosting algorithm.

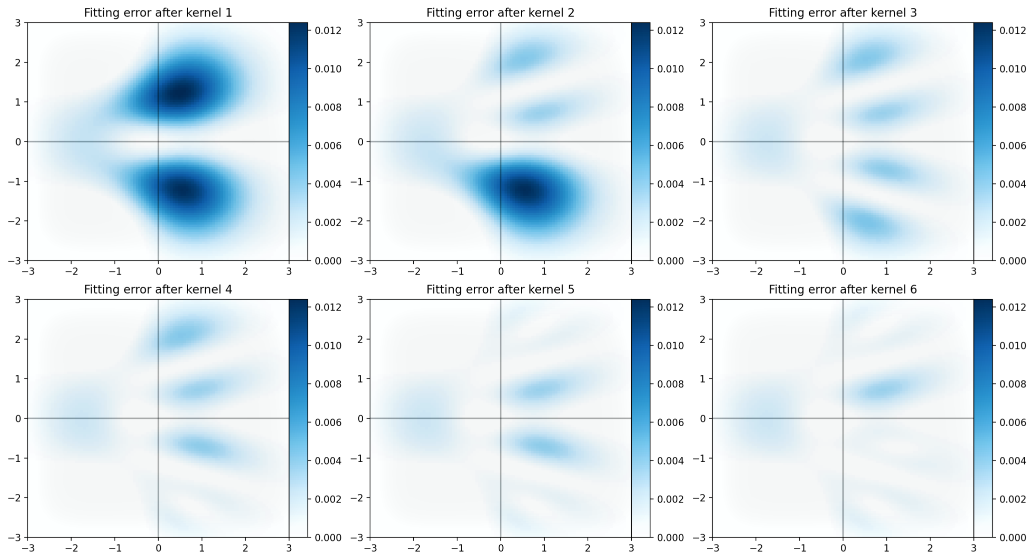

Figure 3: Example 1. Performance of the adaptive boosting algorithm in reducing approximation errors in tting the nonlinear component of the operator.

4.2 Example 2: Bearing-only tracking

In this example, we solve the bearing-only tracking problem, which is a benchmark optimal ltering problem in practice. Specically, we aim to track a moving target driven by the following state dynamics

$$dX_t = \begin{bmatrix} v_t^1 \\ v_t^2 \\ 0 \\ 0 \end{bmatrix} dt + \begin{bmatrix} {}_1 & 0 & 0 & 0 \\ 0 & {}_2 & 0 & 0 \\ 0 & 0 & {}_3 & 0 \\ 0 & 0 & 0 & {}_4 \end{bmatrix} \begin{bmatrix} dW_t^1 \\ dW_t^2 \\ dW_t^3 \\ dW_t^4 \end{bmatrix}; \tag{22}$$

where $X_t = [x_t^1; x_t^2; v_t^1; v_t^2]^{>}$, $[x_t^1; x_t^2]^{>}$ describes the 2D location of the target, and v_t^1, v_t^2 are the velocities in x_1 and x_2 directions, respectively. $W_t = [W_t^1; W_t^2; W_t^3; W_t^4]$ is a 4D Brownian motion that brings uncertainty to the state model, which is driven by the diusion coecient

$$:= \begin{bmatrix} _1 & 0 & 0 & 0 \\ 0 & _2 & 0 & 0 \\ 0 & 0 & _3 & 0 \\ 0 & 0 & 0 & _4 \end{bmatrix}.$$

In order to estimate the location of the target, we place two detectors on dierent observation platforms to collect bearing angles as observational data. Specically, the observational data is given by the following observational function

$$Y_t^i = \arctan \frac{x_t^2 \quad x_{\text{i-platform}}^2}{x_t^1 \quad x_{\text{i-platform}}^1} + {}_i; \qquad i = 1; 2; \tag{23}$$

where $(x_{\text{i-platform}}^1; x_{\text{i-platform}}^2)^{>}$ gives the location of the i-th platform, and $_i$ is the observation noise of the i-th detector.

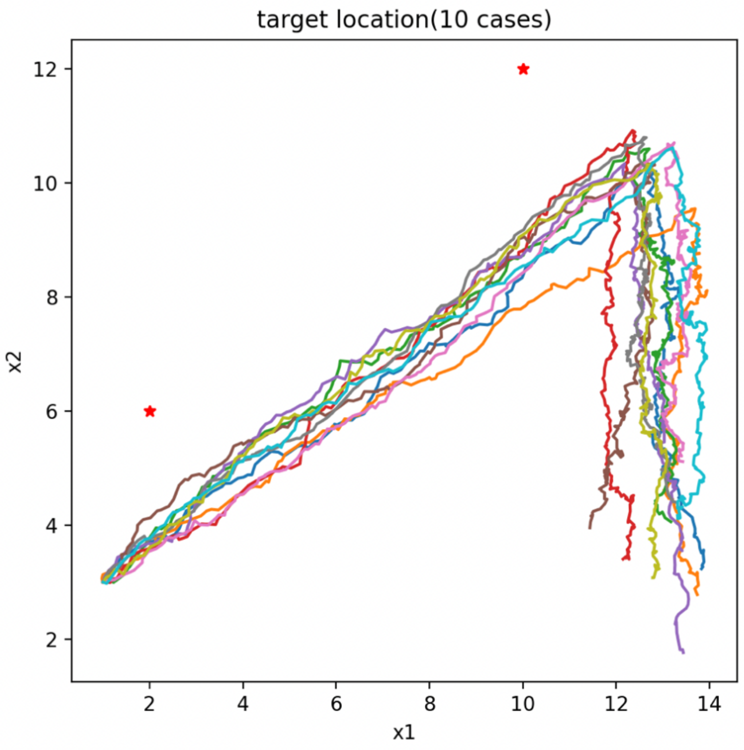

Figure 4: Example 2. Demonstration of 10 sample trajectories of the target.

In this example, we track the target over the time period t 2 [0; 3] with initial state $X_0 = [1; 3; 10; 6]^{>}$, and we let t = 0:01, i.e. we track 300 time steps. The diusion coecient is chosen as $_1 = {}_2 = 0{:}5$, $_3 = {}_4 = 0{:}3$, and we locate two platforms at $(2; 6)^{>}$ and $(10; 12)^{>}$, respectively. To demonstrate the stability of our method compared with other state-of-the-art methods, we assume that there's an unexpected turn in the target moving direction at the time instant t = 1:2, which would challenge the robustness of optimal ltering methods. In Figure 4, we plot 10 sample target trajectories (by using dierent colors), and we mark the observation platforms with red stars. From this gure, we can see that the target is designed to move in front of the observation platforms and then it makes a sharp turn downward.

In Figure 5, we present a comparison experiment, in which we compare the tracking accuracy between our adaptive kernel method with two state-of-the-art optimal ltering methods, i.e. the ensemble Kalman lter and the particle lter. To implement our adaptive kernel method, we use up

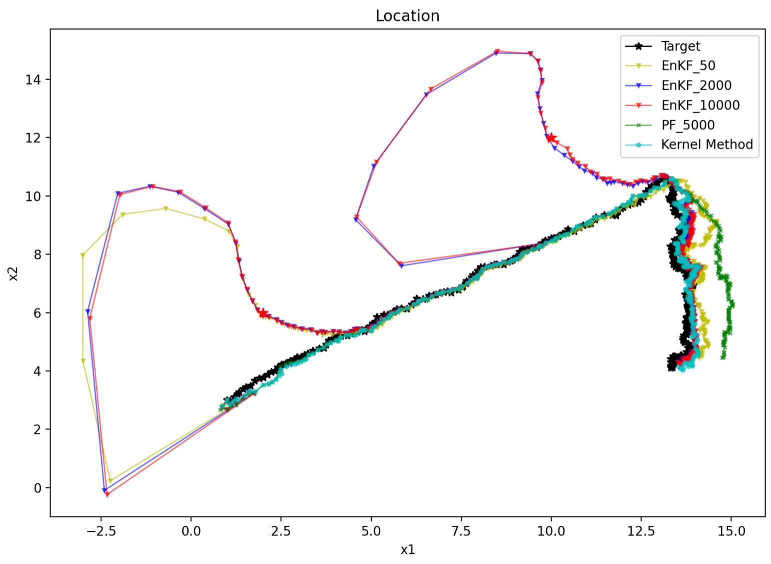

Figure 5: Example 2. Comparison of tracking performance in solving the bearing-only tracking problem.

to 20 kernels to approximate the ltering density, and active kernels are adaptively selected by the boosting algorithm (described in Table 1). For the ensemble Kalman lter, we choose 50, 2; 000 and 10; 000 realizations of Kalman lter samples to implement this tracking task. In the particle lter, we use 5000 particles to generate empirical distributions for the ltering density. In the gure, we use the black curve (marked by stars) to represent a sample of real target trajectory and use other colored curves to represent the estimates obtained by various optimal ltering methods. The yellow, blue, and red curves (marked by triangles) are estimates for the target location obtained by using the ensemble Kalman lter (EnKF) with 50, 2; 000, and 10; 000 realizations of Kalman lter samples, respectively. The green curve (marked by crosses) gives the particle lter (PF) estimates (obtained by using 5; 000 particles). The cyan curve (marked by dots) describes the estimates obtained by our adaptive kernel method.

From this gure, we can see that the EnKF doesn't provide accurate estimates for the target location when the target is right below a detector { no matter how many realizations of samples we use in the EnKF. The poor performance of the EnKF is caused by the high nonlinearity of observational data (bearing angles introduced in Eq. (23)) when the target moves in front of detectors. For the PF, we can see that it provides accurate estimates until the sharp turn at the time instant t = 1:2. Then, the PF loses track of the target due to the degeneracy of particles when trying to adjust the change of the target location. On the other hand, the kernel method always keeps on track, and it gives accurate estimates all the time during the tracking period.

To further examine the performance of dierent optimal ltering methods in solving the bearing-only tracking problem (22)-(23), we repeat the above experiment 100 times and calculate the root mean square errors (RMSEs) of target tracking performance. The log scaled RMSEs of each method with respect to time are presented in Figure 6. From this gure, we can see that the adaptive kernel method (cyan curve marked by dots) has the lowest RMSEs, and it can provide good accuracy even after the sharp turn of the target. The PF (green curve marked by crosses) has low RMSEs at rst. However, the errors increase dramatically at the turning point of the target trajectories. On the other hand, the EnKF estimates (yellow, blue and red curves marked by triangles) always suer from low accuracy when the target passes the detectors. But the EnKF can recover quickly from inaccurate estimates, which indicates that the EnKF is a more stable method compared with the PF.

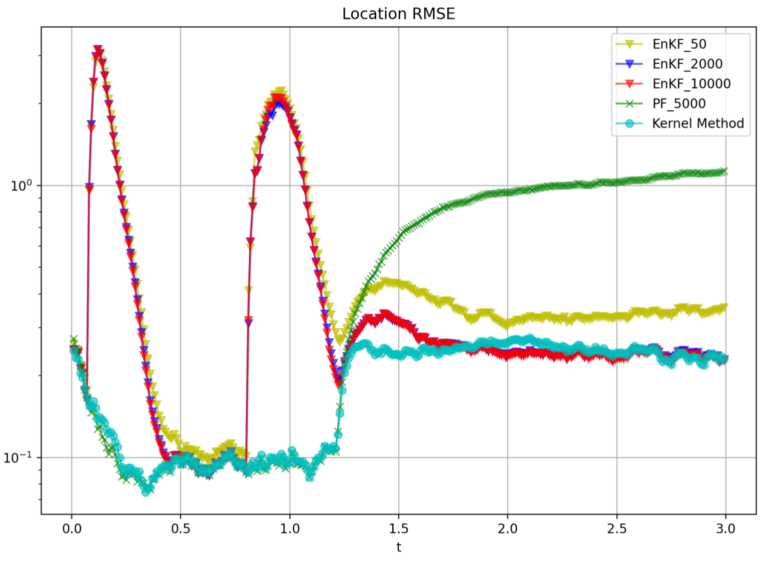

Figure 6: Example 2. Comparison of root mean square errors (RMSEs) with respect to time.

To summarize the general performance of each method, we present the accumulated RMSEs (the combined RMSEs over the tracking period) together with the CPU time of each method (average over the above 100 repeated tests) in Table 3. The CPU that we use is a AMD Ryzen 5 5600X

Table 3: Performance comparison

	EnKF 50	EnKF 2,000	EnKF 10,000	PF 5000	Kernel Method
Accumulated RMSEs	158:99	134:75	134:87	169:58	56:75
CPU time (seconds)	0:32	11	56	70	50

with 6 core 12 processing threads. We can see from this table that the PF has the lowest accuracy with the highest computational cost, which is mainly caused by the degeneracy of particles. The EnKF can solve the problem with very low computational cost. However, the accuracy of the EnKF does not improve much even we use a lot more realizations of Kalman lter samples. The kernel learning method, on the other hand, has much lower RMSEs compared with the EnKF and the PF with moderate cost.

4.3 Example 3: Lorenz-96 model

To examine the performance of the adaptive kernel method in solving high dimensional problems, in this example we solve the Lorenz-96 tracking problem, which is a benchmark high dimensional optimal ltering problem. The state model is given by the following stochastic dynamical system

$$x_t^i = ((x_t^{i+1} \quad x_t^{i\ 2})x_t^{i\ 1} \quad x_t^i + F)dt + {}^i dW_t^i; \qquad i = 1; 2; \ ; d \tag{24}$$

where $X_t = [x_t^1; x_t^2 \ ; x^d]_t^>$ is the target state. In the Lorenz-96 model (24), we let $x\ {}_t^1 = x^d\ {}_t^1$, $x^0 =_t x^d$, $x_t^1 = {}_t x^{d+1}$, ${}_t W_t = fW^1; W_t^2 \ ; W_t^d g$ is a d-dimensional Brownian motion, and $= [{}^1; {}^2 \ ; {}^d]^>$ is the diusion coecient. It is well-known that when $F = 8$, the Lorenz-96 model has chaotic behavior, which makes the corresponding optimal ltering problem very challenging. In this example, we track the state X of the Lorenz-96 model over the time period $t\ 2\ [0; 3]$, and we let $d = 10$. As a commonly used scenario when tracking the Lorenz-96 model, we simulate the Lorenz-96 model with time step-size $t = 0{:}001$, and we assume that we receive data of the state with time step-size $t = 0{:}1$. Therefore, the Bayesian inference procedure is implemented after every 100 simulation steps. In other words, we carry out one update step in every 100 predication steps.

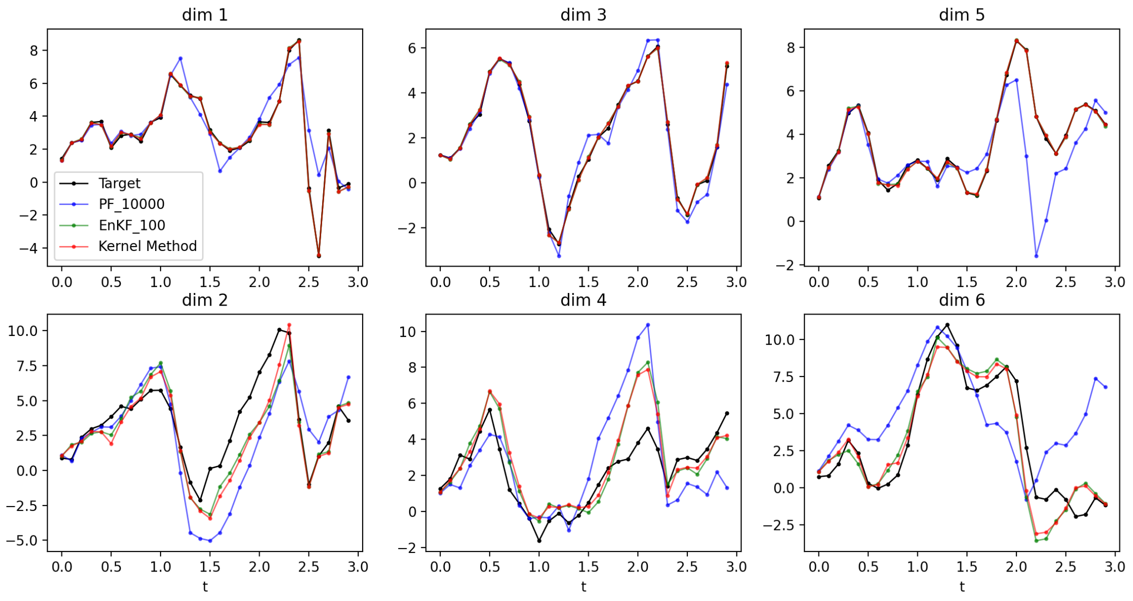

Figure 7: Example 3. Comparison of tracking performance in dimensions 1 to 6.

In this example, the observational data that we receive to estimate the state of the Lorenz-96 model are noise perturbed direct state observations in odd dimensions, i.e.

$$Y_t = [x_t^1; x_t^3; x_t^5; x_t^7; x_t^9]^> + {}_t;$$

where $_t$ N(0; 1) is a standard Gaussian noise.

In Figure 7, we compare our kernel method with the PF and the EnKF, and we present the state estimation performance of each method in the rst six dimensions. In each subplot, the black curve shows a sample of real target trajectory of the Lorenz-96 model state. The blue curve is the PF estimates obtained by using 10; 000 particles to represent the empirical distribution of the state. The green curve is the EnKF estimates obtained by using 100 realizations of Kalman lter samples. The red curve gives the estimates obtained by our kernel method, and we use at most 20 kernels to approximate the ltering density in the adaptive boosting algorithm when tting the nonlinear component of the state drift. From this gure, we can see that the EnKF has comparable estimation performance to the kernel method due to the linear observations, and the usage of \ensemble estimation" in the EnKF can handle the nonlinearity of the state dynamics. On the other hand, the PF provides low tracking accuracy. Especially, the long simulation period (without an update) in this example would cause more severe degeneracy issue since no data can be used to resample the particles.

To conrm the comparison result presented in Figure 7, we repeat the above experiment 100 times and present the log scaled RMSEs of each method with respect to time in Figure 8. We can see from this gure that the PF has much higher errors compared with the EnKF and the kernel method while both the EnKF and the kernel method have similar RMSEs in this Lorenz-96 tracking problem.

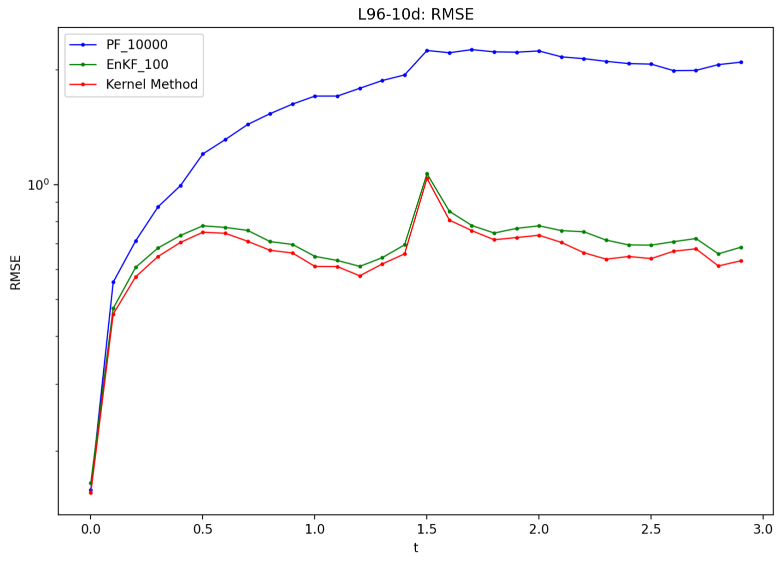

Figure 8: Example 3. Comparison of RMSEs with respect to time.

5 Summary and conclusions

In this paper, we developed an adaptive kernel method to solve the optimal ltering problem. The main idea of our method is to use a set of Gaussian kernels to approximate the ltering density of a target dynamical state model. Due to the fact that the ltering density describes a probabilistic distribution, Gaussian kernels can eectively characterize the distribution, which is often a bell-shaped function. Then, an operator decomposition method is introduced to eciently propagate the state of the model, and adaptive boosting algorithm is applied to adaptively capture important features of the ltering density.

Three numerical experiments are presented to examine the performance of our kernel method. In the rst example, we presented the eectiveness of the adaptive kernel method in characterizing propagation of the ltering density. In the second example and the third example, we compared the performance of the kernel method with two state-of-the-art methods, i.e. the particle lter and the ensemble Kalman lter, in solving benchmark optimal ltering problems. Results in our numerical experiments indicate that our method has high accuracy and high stability advantage compared with the particle lter, and it outperforms the ensemble Kalman lter when data provide highly nonlinear state observations.

Acknowledgement

This work is partially supported by U.S. Department of Energy through FASTMath Institute and Oce of Science, Advanced Scientic Computing Research program under the grant DE-SC0022297. The second author (FB) would also like to acknowledge the support from U.S. National Science Foundation through project DMS-2142672.

References

[1] C. Andrieu, A. Doucet, and R. Holenstein. Particle markov chain monte carlo methods. J. R. Statist. Soc. B, 72(3):269{342, 2010.

[2] R. Archibald and F. Bao. A kernel learning method for backward sde lter. arXiv: 2201.10600, 1, 2021.

[3] F. Bao, Y. Cao, and H. Chi. Adjoint forward backward stochastic dierential equations driven by jump diusion processes and its application to nonlinear ltering problems. Int. J. Uncertain. Quantif., 9(2):143{159, 2019.

[4] F. Bao, Y. Cao, and X. Han. Forward backward doubly stochastic dierential equations and optimal ltering of diusion processes. Communications in Mathematical Sciences, 18(3):635{ 661, 2020.

[5] F. Bao, Y. Cao, C. Webster, and G. Zhang. A hybrid sparse-grid approach for nonlinear ltering problems based on adaptive-domain of the Zakai equation approximations. SIAM/ASA J. Uncertain. Quantif., 2(1):784{804, 2014.

[6] F. Bao, Y. Cao, and W. Zhao. Numerical solutions for forward backward doubly stochastic dierential equations and zakai equations. International Journal for Uncertainty Quantication, 1(4):351{367, 2011.

[7] F. Bao, Y. Cao, and W. Zhao. A rst order semi-discrete algorithm for backward doubly stochas-tic dierential equations. Discrete and Continuous Dynamical Systems-Series B, 5(2):1297 { 1313, 2015.

[8] F. Bao, Y. Cao, and W. Zhao. A backward doubly stochastic dierential equation approach for nonlinear ltering problems. Commun. Comput. Phys., 23(5):1573{1601, 2018.

[9] F. Bao and V. Maroulas. Adaptive meshfree backward SDE lter. SIAM J. Sci. Comput., 39(6):A2664{A2683, 2017.

[10] A. Davie and J. Gaines. Convergence of numerical schemes for the solution of the parabolic stochastic partial dierential equations. Math. Comp., 70:121{134, 2001.

[11] N. El Karoui, S. Peng, and M. C. Quenez. Backward stochastic dierential equations in nance. Math. Finance, 7(1):1{71, 1997.

[12] G. Evensen. Data assimilation: the ensemble Kalman lter. Springer, 2006.

[13] G. Evensen. The ensemble Kalman lter for combined state and parameter estimation: Monte Carlo techniques for data assimilation in large systems. IEEE Control Syst. Mag., 29(3):83{104, 2009.

[14] Emmanuel Gobet, Gilles Pages, Huyên Pham, and Jacques Printems. Discretization and simulation of the Zakai equation. SIAM J. Numer. Anal., 44(6):2505{2538 (electronic), 2006.

[15] Mehmet G•nen and Ethem Alpayd n. Multiple kernel learning algorithms. J. Mach. Learn. Res., 12:2211{2268, 2011.

[16] N.J Gordon, D.J Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. IEE PROCEEDING-F, 140(2):107{113, 1993.

[17] W. Grecksch and P. E. Kloeden. Time-discretised Galerkin approximations of parabolic stochastic PDEs. Bull. Austral. Math. Soc., 54(1):79{85, 1996.

[18] Thomas Hofmann, Bernhard Sch•lkopf, and Alexander J. Smola. Kernel methods in machine learning. Ann. Statist., 36(3):1171{1220, 2008.

[19] Sun Hui and Feng Bao. Meshfree approximation for stochastic optimal control problems. Communications in Mathematical Research, 37(3):387{420, 2021.

[20] R. E. Kalman and R. S. Bucy. New results in linear ltering and prediction theory. Transactions of the ASME{Journal of Basic Engineering, 83(Series D):95{108, 1961.

[21] K. Kang, V. Maroulas, I. Schizas, and F. Bao. Improved distributed particle lters for tracking in a wireless sensor network. Comput. Statist. Data Anal., 117:90{108, 2018.

[22] H.J. Kushner and P. Dupuis. Numerical methods for stochastic control problems in continuous time. In Applications of Mathematics, volume 24. Springer-Verlag, New York, 1992.

[23] Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. Boosting algorithms as gradient descent. In S. Solla, T. Leen, and K. M•ller, editors, Advances in Neural Information Processing Systems, volume 12. MIT Press, 2000.

[24] Etienne Pardoux and Shi Ge Peng. Backward doubly stochastic dierential equations and systems of quasilinear SPDEs. Probab. Theory Related Fields, 98(2):209{227, 1994.

[25] Michael K. Pitt and Neil Shephard. Filtering via simulation: auxiliary particle lters. J. Amer. Statist. Assoc., 94(446):590{599, 1999.

[26] M. J. D. Powell. Radial Basis Functions for Multivariable Interpolation: A Review, page 143{167. Clarendon Press, USA, 1987.

[27] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. Obstacles to high-dimensional particle ltering. Mon. Wea. Rev., 136:4629{4640, 2008.

[28] Dougal J. Sutherland and Je Schneider. On the error of random fourier features. UAI'15, Arlington, Virginia, USA, 2015. AUAI Press.

[29] P. J. van Leeuwen. Nonlinear data assimilation in geosciences: an extremely ecient particle lter. Q. J. Roy. Meteor. Soc., 136(653):1991{1999, 2010.

[30] Bin Wang, Xiaolei Zou, and Jiang Zhu. Data assimilation and its applications. Proceedings of the National Academy of Sciences, 97(21):11143{11144, 2000.

[31] Dongbin Xiu and Jan S. Hesthaven. High-order collocation methods for dierential equations with random inputs. SIAM J. Sci. Comput., 27(3):1118{1139 (electronic), 2005.

[32] Moshe Zakai. On the optimal ltering of diusion processes. Z. Wahrscheinlichkeitstheorie und Verw. Gebiete, 11:230{243, 1969.

[33] Guannan Zhang and Max Gunzburger. Error analysis of a stochastic collocation method for parabolic partial dierential equations with random input data. SIAM J. Numer. Anal., 50(4):1922{1940, 2012.

[34] H. Zhang and D. Laneuville. Grid based solution of zakai equation with adaptive local renement for bearing-only tracking. IEEE Aerospace Conference, 2008.