Parameter Estimation with Data-Driven Nonparametric Likelihood Functions

Shixiao W. Jiang [1,†] and **John Harlim** [1,2,3,*,†]

1   Department of Mathematics, the Pennsylvania State University, 109 McAllister Building, University Park, PA 16802-6400, USA; suj235@psu.edu
2   Department of Meteorology and Atmospheric Science, the Pennsylvania State University, 503 Walker Building, University Park, PA 16802-5013, USA
3   Institute for CyberScience, the Pennsylvania State University, 224B Computer Building, University Park, PA 16802, USA
*   Correspondence: jharlim@psu.edu
†   These authors contributed equally to this work.

Abstract: In this paper, we consider a surrogate modeling approach using a data-driven nonparametric likelihood function constructed on a manifold on which the data lie (or to which they are close). The proposed method represents the likelihood function using a spectral expansion formulation known as the kernel embedding of the conditional distribution. To respect the geometry of the data, we employ this spectral expansion using a set of data-driven basis functions obtained from the diffusion maps algorithm. The theoretical error estimate suggests that the error bound of the approximate data-driven likelihood function is independent of the variance of the basis functions, which allows us to determine the amount of training data for accurate likelihood function estimations. Supporting numerical results to demonstrate the robustness of the data-driven likelihood functions for parameter estimation are given on instructive examples involving stochastic and deterministic differential equations. When the dimension of the data manifold is strictly less than the dimension of the ambient space, we found that the proposed approach (which does not require the knowledge of the data manifold) is superior compared to likelihood functions constructed using standard parametric basis functions defined on the ambient coordinates. In an example where the data manifold is not smooth and unknown, the proposed method is more robust compared to an existing polynomial chaos surrogate model which assumes a parametric likelihood, the non-intrusive spectral projection. In fact, the estimation accuracy is comparable to direct MCMC estimates with only eight likelihood function evaluations that can be done offline as opposed to 4000 sequential function evaluations, whenever direct MCMC can be performed. A robust accurate estimation is also found using a likelihood function trained on statistical averages of the chaotic 40-dimensional Lorenz-96 model on a wide parameter domain.

Keywords: Bayesian inference; MCMC; diffusion maps; nonparametric likelihood function; surrogate modeling; reproducing kernel Hilbert space; kernel embedding of the conditional distribution

1. Introduction

Bayesian inference is a popular approach for solving inverse problems with far-reaching applications, such as parameter estimation and uncertainty quantification (see for example [1–3]). In this article, we will focus on a classical Bayesian inference problem of estimating the conditional distribution of hidden parameters of dynamical systems from a given set of noisy observations.

In particular, let $\mathbf{x}(t; \boldsymbol{\theta})$ be a time-dependent state variable, which implicitly depends on the parameter $\boldsymbol{\theta}$ through the following initial value problem,

$$\dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{\theta}), \quad \mathbf{x}(0) = \mathbf{x}_0. \tag{1}$$

Here, for any fixed $\boldsymbol{\theta}$, f can be either deterministic or stochastic. Our goal is to estimate the conditional distribution of $\boldsymbol{\theta}$, given discrete-time noisy observations $\mathbf{y}^\dagger = \{\mathbf{y}_1^\dagger, \dots, \mathbf{y}_T^\dagger\}$, where:

$$\mathbf{y}_i^\dagger = g(\mathbf{x}_i^\dagger, \boldsymbol{\xi}_i), \quad i = 1, \dots, T. \tag{2}$$

Here, $\mathbf{x}_i^\dagger \equiv \mathbf{x}(t_i; \boldsymbol{\theta}^\dagger)$ are the solutions of Equation (1) for a specific hidden parameter $\boldsymbol{\theta}^\dagger$, g is the observation function, and $\boldsymbol{\xi}_i$ are unbiased noises representing the measurement or model error. Although the proposed approach can also estimate the conditional density of the initial condition $\mathbf{x}_0$, we will not explore this inference problem in this article.

Given a prior density, $p_0(\boldsymbol{\theta})$, Bayes' theorem states that the conditional distribution of the parameter $\boldsymbol{\theta}$ can be estimated as,

$$p(\boldsymbol{\theta}|\mathbf{y}^\dagger) \propto p(\mathbf{y}^\dagger|\boldsymbol{\theta}) p_0(\boldsymbol{\theta}), \tag{3}$$

where $p(\mathbf{y}^\dagger|\boldsymbol{\theta})$ denotes the likelihood function of $\boldsymbol{\theta}$ given the measurements $\mathbf{y}^\dagger$ that depend on a hidden parameter value $\boldsymbol{\theta}^\dagger$ through (2). In most applications, the statistics of the conditional distribution $p(\boldsymbol{\theta}|\mathbf{y}^\dagger)$ are the quantity of interest. For example, one can use the mean statistic as a point estimator of $\boldsymbol{\theta}^\dagger$ and the higher order moments for uncertainty quantification. To realize this goal, one draws samples of $p(\boldsymbol{\theta}|\mathbf{y}^\dagger)$ and estimates these statistics via Monte Carlo averages over these samples. In this application, Markov Chain Monte Carlo (MCMC) is a natural sampling method that plays a central role in the computational statistics behind most Bayesian inference techniques [4].

In our setup, we assume that for any $\boldsymbol{\theta}$, one can simulate:

$$\mathbf{y}_i(\boldsymbol{\theta}) = g(\mathbf{x}_i(\boldsymbol{\theta}), \boldsymbol{\xi}_i), i = 1, \dots, T. \tag{4}$$

where $\mathbf{x}_i(\boldsymbol{\theta}) \equiv \mathbf{x}(t_i; \boldsymbol{\theta})$ denote solutions to the initial value problem in Equation (1). If the observation function has the following form,

$$g(\mathbf{x}_i(\boldsymbol{\theta}), \boldsymbol{\xi}_i) = h(\mathbf{x}_i(\boldsymbol{\theta})) + \boldsymbol{\xi}_i, \tag{5}$$

where $\boldsymbol{\xi}_i$ are i.i.d. noises, then one can define the likelihood function of $\boldsymbol{\theta}$, $p(\mathbf{y}^\dagger|\boldsymbol{\theta})$, as a product of the density functions of the noises $\boldsymbol{\xi}_i$,

$$p(\mathbf{y}^\dagger|\boldsymbol{\theta}) \equiv \prod_{i=1}^{T} p(\boldsymbol{\xi}_i) = \prod_{i=1}^{T} p(\mathbf{y}_i^\dagger - h(\mathbf{x}_i(\boldsymbol{\theta}))). \tag{6}$$

When the observations are noise-less, $\boldsymbol{\xi}_i = 0$, and the underlying system is an Itô diffusion process with additive or multiplicative noises, one can use the Bayesian imputation to approximate the likelihood function [5]. In both parametric approaches, it is worth noting that the dependence of the likelihood function on the parameter is implicit through the solutions $\mathbf{x}_i(\boldsymbol{\theta})$. Practically, this implicit dependence is the source of the computational burden in evaluating the likelihood function since it requires solving the dynamical model in (1) or every proposal in the MCMC chain. In the case when simulating $\mathbf{y}_i(\boldsymbol{\theta})$ is computationally feasible, but the likelihood function is intractable, then one can use, e.g., the Approximate Bayesian Computation (ABC) rejection algorithm [6,7] for Bayesian inference. Basically, the ABC rejection scheme generates the samples of $p(\boldsymbol{\theta}|\mathbf{y}^\dagger)$ by comparing the simulated $\mathbf{y}_i(\boldsymbol{\theta})$ to the observed data, $\mathbf{y}_i^\dagger$, with an appropriate choice of metric comparison for each proposal $\boldsymbol{\theta} \sim p_0(\boldsymbol{\theta})$. In general, however, repetitive evaluation of (4) can be expensive when the dynamics in (1) is high-dimensional and/or stiff, or when T is large, or when the function g is an average of

a long time series. Our goal is to address this situation in addition to not knowing the approximate likelihood function.

Broadly speaking, the existing approaches to overcome repetitive evaluation of (4) require knowledge of an approximate likelihood function such as in (6). They can be grouped into two classes. The first class consists of methods that improve/accelerate the sampling strategy; for example, the Hamiltonian Monte Carlo [8], adaptive MCMC [9], and delay rejection adaptive Metropolis [10], just to name a few. The second class consists of methods that avoid solving the dynamical model in (1) when running the MCMC chain by replacing it with a computationally more efficient model on a known parameter domain. This class of approach, also known as *surrogate modeling*, includes Gaussian process models [11], polynomial chaos [12,13], and enhanced model error [14]; for example, the non-intrusive spectral projection [13] approximate $\mathbf{x}_i(\boldsymbol{\theta})$ in (6) with a polynomial chaos expansion. Another related approach, which also avoids MCMC on top of integrating (1), is to employ a polynomial expansion on the likelihood function [15]. This method represents the parametric likelihood function in (6) with orthonormal basis functions of a Hilbert space weighted by the prior measure. This choice of basis functions makes the computation for the statistics of the posterior density straightforward, and thus, MCMC is not needed.

In this paper, we consider a surrogate modeling approach where a nonparametric likelihood function is constructed using a data-driven spectral expansion. By nonparametric, we mean that our approach does not require any parametric form or assume any distribution as in (6). Instead, we approximate the likelihood function using the kernel embedding of conditional distribution formulation introduced in [16,17]. In our application, we will extend their formulation onto a Hilbert space weighted by the sampling measure of the training dataset as in [18]. We will rigorously demonstrate that using orthonormal basis functions of this data-driven weighted Hilbert space, the error bound is independent of the variance of the basis functions, which allows us to determine the amount of training data for accurate likelihood function estimations.

Computationally, assuming that the observations lie on (or close to) a Riemannian manifold $\mathcal{N}$ embedded in $\mathbb{R}^n$ with sampling density $q(\mathbf{y})$, we apply the diffusion maps algorithm [19,20] to approximate orthonormal basis functions $\varphi_k \in L^2(\mathcal{N}, q)$ using the training dataset. Subsequently, a nonparametric likelihood function is represented as a weighted sum of these data-driven basis functions, where the coefficients are precomputed using the kernel embedding formulation. In this fashion, our approach respects the geometry of the data manifold. Using this nonparametric likelihood function, we then generate the MCMC chain for estimating the conditional distribution of hidden parameters. For the present work, our aim is to demonstrate that one can obtain accurate and robust parameter estimation by implementing a simple Bayesian inference algorithm, the Metropolis scheme, with the data-driven nonparametric likelihood function. We should also point out that the present method is computationally feasible on low-dimensional parameter space, like any other surrogate modeling approach. Possible ways to overcome this dimensionality issue will be discussed.

This paper is organized as follows: In Section 2, we review the formulation of the reproducing kernel Hilbert space to estimate conditional density functions. In Section 3, we discuss the error estimate of the likelihood function approximation. In Section 4, we discuss the construction of the analytic basis functions for the Euclidean data manifold, as well as the data-driven basis functions with the diffusion maps algorithm for data that lie on embedded Riemannian geometry. In Section 5, we provide numerical results with parameter estimation application on instructive examples. In one of the examples where the dynamical model is low-dimensional and the observation is in the form of (5), we compare the proposed approach with the direct MCMC and non-intrusive spectral projection method (both schemes use likelihood of the form (6)). In addition, we will also demonstrate the robustness of the proposed approach on an example where g is a statistical average of a long-time trajectory (in which the likelihood is intractable) and the dynamical model has relatively high-dimensional chaotic dynamics such that repetitive evaluation of (4) is numerically expensive. In

Section 6, we conclude this paper with a short summary. We accompany this paper with Appendices for treating large amount of data and more numerical results.

2. Conditional Density Estimation via Reproducing Kernel Weighted Hilbert Spaces

Let $\mathbf{y} \in \mathcal{N} \subseteq \mathbb{R}^n$, where $\mathcal{N}$ is a smooth manifold with intrinsic dimension $d \leq n$. In practice, we measure the observations in the ambient coordinates and denote their components as $\mathbf{y} = \{y^1, \ldots, y^n\}$. For the parameter $\boldsymbol{\theta}$ space, $\mathcal{M}$ has a Euclidean structure with components, $\boldsymbol{\theta} = \{\theta^1, \ldots, \theta^m\}$, so $\mathcal{M}$ is assumed to be either an m-dimensional hyperrectangle or $\mathbb{R}^m$. For training, we are given M number of training parameters $\{\boldsymbol{\theta}_j\}_{j=1,\ldots,M} = \{\theta^1_j, \ldots, \theta^m_j\}_{j=1,\ldots,M}$. For each training parameter $\boldsymbol{\theta}_j$, we generate a discrete time series of length N for noisy observation data $\{\mathbf{y}_{i,j}\} = \{y^1_{i,j}, \ldots, y^n_{i,j}\} \in \mathbb{R}^n$ for $i = 1, \ldots, N$, and $j = 1, \ldots, M$. Here, the sub-index i and the sub-index j of $\mathbf{y}_{i,j}$ correspond to the ith observation data for the jth training parameter $\boldsymbol{\theta}_j$. Our goal for training is to learn the conditional density $p(\mathbf{y}|\boldsymbol{\theta})$ from the training dataset $\{\boldsymbol{\theta}_j\}_{j=1,\ldots,M}$ and $\{\mathbf{y}_{i,j}\}_{j=1,\ldots,M}^{i=1,\ldots,N}$ for arbitrary $\mathbf{y}$ and $\boldsymbol{\theta}$ within the range of $\{\boldsymbol{\theta}_j\}_{j=1,\ldots,M}$.

The construction of the conditional density $p(\mathbf{y}|\boldsymbol{\theta})$ is based on a machine learning tool known as the kernel embedding of the conditional distribution formulation introduced in [16,17]. In their formulation, the representation of conditional distributions is an element of a Reproducing Kernel Hilbert Space (RKHS).

Recently, the representation using a Reproducing Kernel Weighted Hilbert Space (RKWHS) was introduced in [18]. That is, let $\Psi_k := \psi_k q$ be the orthonormal basis of $L^2(\mathcal{N}, q^{-1})$, where they are eigenbasis of an integral operator,

$$\mathcal{K}f(\mathbf{y}) = \int_{\mathcal{N}} K(\mathbf{y}, \mathbf{y}') f(\mathbf{y}') q^{-1}(\mathbf{y}') dV, \quad f \in L^2(\mathcal{N}, q^{-1}), \tag{7}$$

that is, $\mathcal{K}\Psi_k = \lambda_k \Psi_k$.

In the case where $\mathcal{N}$ is compact and $\mathcal{K}$ is Hilbert–Schmidt, the kernel can be written as,

$$K(\mathbf{y}, \mathbf{y}') = \sum_{k=1}^{\infty} \lambda_k \Psi_k(\mathbf{y}) \Psi_k(\mathbf{y}'), \tag{8}$$

which converges in $L^2(\mathcal{N}, q^{-1})$. Define the feature map $\Phi : \mathcal{M} \to \ell_2$ as,

$$\Phi(\mathbf{y}) := \{\Phi_k(\mathbf{y}) = \sqrt{\lambda_k} \Psi_k(\mathbf{y}) : k \in \mathbb{Z}^+, \mathbf{y} \in \mathcal{N}\}. \tag{9}$$

Therefore, any $f \in L^2(\mathcal{N}, q^{-1})$ can be represented as $f = \sum_{k=1}^{\infty} \hat{f}_k \Psi_k = \sum_{k=1}^{\infty} \frac{\hat{f}_k}{\sqrt{\lambda_k}} \Phi_k$, where $\hat{f}_k = \langle f, \Psi_k \rangle_{q^{-1}} = \langle f, \psi_k \rangle := \int_{\mathcal{N}} f(\mathbf{y}) \psi_k(\mathbf{y}) dV$ and provided that $\sum_k |\hat{f}_k|^2 / \lambda_k < \infty$. If we define $\langle f, g \rangle_{\mathcal{H}_{q^{-1}}} := \sum_{k=1}^{\infty} \frac{\hat{f}_k \hat{g}_k}{\lambda_k}$, we can write the kernel in (8) as $K(\mathbf{y}, \mathbf{y}') = \langle \Phi(\mathbf{y}), \Phi(\mathbf{y}') \rangle_{\mathcal{H}_{q^{-1}}}$. Throughout this manuscript, we denote the RKHS $\mathcal{H}_{q^{-1}}(\mathcal{N})$ generating the feature map Φ in (9) as the space of square integrable functions with a reproducing property,

$$f(\mathbf{y}) = \langle f, K(\cdot, \mathbf{y}) \rangle_{\mathcal{H}_{q^{-1}}} := \sum_{k=1}^{\infty} \frac{\hat{f}_k \langle K(\cdot, \mathbf{y}), \Psi_k \rangle_{q^{-1}}}{\lambda_k} = \sum_{k=1}^{\infty} \frac{\hat{f}_k}{\sqrt{\lambda_k}} \Phi_k(\mathbf{y}) = \langle f, \Phi(\mathbf{y}) \rangle_{\mathcal{H}_{q^{-1}}}, \quad \forall \mathbf{y} \in \mathcal{N},$$

induced by the basis of $\Psi_k \in L^2(\mathcal{N}, q^{-1})$. While this definition deceptively suggests that $\mathcal{H}_{q^{-1}}(\mathcal{N})$ is similar to $L^2(\mathcal{N}, q^{-1})$, we should also point out that the RKHS requires that the Dirac functional $\delta_x : \mathcal{H}_{q^{-1}}(\mathcal{N}) \to \mathbb{R}$ defined as $\delta_x f = f(x)$ be continuous. Since L^2 contains a class of functions, it is not an RKHS and $\mathcal{H}_{q^{-1}}(\mathcal{N}) \subset L^2(\mathcal{N}, q^{-1})$. See, e.g., Chapter 4 of [21] for more details. Using the same definition, we denote $\mathcal{H}_{\tilde{q}^{-1}}(\mathcal{M})$ as the RKHS induced by orthonormal basis of $L^2(\mathcal{M}, \tilde{q}^{-1})$ of functions of the parameter $\boldsymbol{\theta}$.

In this work, we will represent conditional density functions using the RKWHS induced by the data, where the bases will be constructed using the diffusion maps algorithm. The outcome of the training is an estimate of the conditional density, $\widehat{p}(\mathbf{y}|\boldsymbol{\theta})$, for arbitrary $\mathbf{y}$ and $\boldsymbol{\theta}$ within the range of $\{\boldsymbol{\theta}_j\}_{j=1,\dots,M}$.

2.1. Review of Nonparametric RKWHS Representation of Conditional Density Functions

We first review the RKWHS representation of conditional density functions deduced in [18]. Let $\psi_k(\mathbf{y})$ be the orthonormal basis functions of $L^2(\mathcal{N}, q)$, where $\mathcal{N}$ contains the domain of the training data $\mathbf{y}_{i,j}$, and the weight function $q(\mathbf{y})$ is defined with respect to the volume form inherited by $\mathcal{N}$ from the ambient space $\mathbb{R}^n$. Let $\varphi_l(\boldsymbol{\theta}) \in L^2(\mathcal{M}, \widetilde{q})$ be the orthonormal basis functions in the parameter $\boldsymbol{\theta}$ space, where the training parameters are $\boldsymbol{\theta}_j \in \mathcal{M}$, with weight function $\widetilde{q}(\boldsymbol{\theta})$. For finite modes, $k = 1, \dots, K_1$, and $l = 1, \dots, K_2$, a nonparametric RKWHS representation of the conditional density can be written as follows [18]:

$$\widehat{p}(\mathbf{y}|\boldsymbol{\theta}) = \sum_{k=1}^{K_1} \widehat{c}_{\mathbf{Y}|\boldsymbol{\theta},k} \psi_k(\mathbf{y}) q(\mathbf{y}), \tag{10}$$

where $\widehat{p}(\mathbf{y}|\boldsymbol{\theta})$ denotes an estimate of the conditional density $p(\mathbf{y}|\boldsymbol{\theta}) \in \mathcal{H}_{q^{-1}}(\mathcal{N})$, and the expansion coefficients are defined as:

$$\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta},k} = \sum_{l=1}^{K_2} \left[\mathbf{C}_{\mathbf{Y}\boldsymbol{\Theta}} \mathbf{C}_{\boldsymbol{\Theta}\boldsymbol{\Theta}}^{-1} \right]_{kl} \varphi_l(\boldsymbol{\theta}). \tag{11}$$

Here, the matrix $\mathbf{C}_{\mathbf{Y}\boldsymbol{\Theta}}$ is $K_1 \times K_2$, and the matrix $\mathbf{C}_{\boldsymbol{\Theta}\boldsymbol{\Theta}}$ is $K_2 \times K_2$, whose components can be approximated by Monte Carlo averages [18]:

$$[\mathbf{C}_{\mathbf{Y}\boldsymbol{\Theta}}]_{ks} = \mathbb{E}_{\mathbf{Y}\boldsymbol{\Theta}}[\psi_k \varphi_s] \approx \frac{1}{MN} \sum_{j=1}^{M} \sum_{i=1}^{N} \psi_k(\mathbf{y}_{i,j}) \varphi_s(\boldsymbol{\theta}_j), \tag{12}$$

$$[\mathbf{C}_{\boldsymbol{\Theta}\boldsymbol{\Theta}}]_{sl} = \mathbb{E}_{\boldsymbol{\Theta}\boldsymbol{\Theta}}[\varphi_s \varphi_l] \approx \frac{1}{M} \sum_{j=1}^{M} \varphi_s(\boldsymbol{\theta}_j) \varphi_l(\boldsymbol{\theta}_j), \tag{13}$$

where the expectations $\mathbb{E}$ are taken with respect to the sampling densities of the training dataset $\{\mathbf{y}_{i,j}\}_{j=1,\dots,M}^{i=1,\dots,N}$ and $\{\boldsymbol{\theta}_j\}_{j=1,\dots,M}$. The equation for the expansion coefficients in Equation (11) is based on the theory of kernel embedding of the conditional distribution [16–18]. See [18] for the detailed proof of Equations (11)–(13). Note that for RKWHS representation, the weight functions q and $\widetilde{q}$ can be different from the sampling densities of the training dataset $\{\mathbf{y}_{i,j}\}_{j=1,\dots,M}^{i=1,\dots,N}$ and $\{\boldsymbol{\theta}_j\}_{j=1,\dots,M}$, respectively. This generalizes the representation in [18], which sets the weights q and $\widetilde{q}$ to be the sampling densities of the training dataset $\{\mathbf{y}_{i,j}\}$ and $\{\boldsymbol{\theta}_j\}$, respectively. If the assumption of $p(\mathbf{y}|\boldsymbol{\theta}) \in \mathcal{H}_{q^{-1}}(\mathcal{N})$ is not satisfied, then $\mathbf{C}_{\boldsymbol{\Theta}\boldsymbol{\Theta}}$ can be singular. In such a case, one can follow the suggestion in [16,17] to regularize the linear regression in (11) by replacing $\mathbf{C}_{\boldsymbol{\Theta}\boldsymbol{\Theta}}^{-1}$ with $(\mathbf{C}_{\boldsymbol{\Theta}\boldsymbol{\Theta}} + \lambda I_{K_2})^{-1}$, where $\lambda \in \mathbb{R}$ is an empirically-chosen parameter and I_{K_2} denotes an identity matrix of size $K_2 \times K_2$.

Incidentally, it is worth mentioning that the conditional density in (10) and (11) is represented as a regression in infinite-dimensional spaces with basis functions $\psi_k(\mathbf{y})$ and $\varphi_l(\boldsymbol{\theta})$. The expression (10) is a nonparametric representation in the sense that we do not assume any particular distribution for the density function $p(\mathbf{y}|\boldsymbol{\theta})$. In this representation, only training dataset $\{\mathbf{y}_{i,j}\}_{j=1,\dots,M}^{i=1,\dots,N}$ and $\{\boldsymbol{\theta}_j\}_{j=1,\dots,M}$ with appropriate basis functions are used to specify the coefficients $\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta},k}$ and the densities $\widehat{p}(\mathbf{y}|\boldsymbol{\theta})$. In Section 4, we will demonstrate how to construct the appropriate basis completely from the training data, motivated by the theoretical result in Section 3 below.

2.2. Simplification of the Expansion Coefficients (11)

If the weight function $\widetilde{q}(\boldsymbol{\theta})$ is the sampling density of the training parameters $\{\boldsymbol{\theta}_j\}_{j=1,\ldots,M}$, the matrix $\mathbf{C}_{\boldsymbol{\Theta\Theta}}$ in (13) can be simplified to a $K_2 \times K_2$ identity matrix,

$$[\mathbf{C}_{\boldsymbol{\Theta\Theta}}]_{sl} = \mathbb{E}_{\boldsymbol{\Theta\Theta}}[\varphi_s \varphi_l] = \int_{\mathcal{M}} \varphi_s(\boldsymbol{\theta}) \varphi_l(\boldsymbol{\theta}) \widetilde{q}(\boldsymbol{\theta}) d\boldsymbol{\theta} = \delta_{sl}. \tag{14}$$

where δ_{sl} is the Kronecker delta function. Here, the second equality follows from the weight $\widetilde{q}(\boldsymbol{\theta})$ being the sampling density, and the third equality follows from the orthonormality of $\varphi_l(\boldsymbol{\theta}) \in L^2(\mathcal{M}, \widetilde{q})$ with respect to the weight function $\widetilde{q}$. Then, the expansion coefficients $\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta},k}$ in (11) can be simplified to,

$$\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta},k} = \sum_{l=1}^{K_2} [\mathbf{C}_{\mathbf{Y}\boldsymbol{\Theta}}]_{kl} \, \varphi_l(\boldsymbol{\theta}), \tag{15}$$

with the $K_1 \times K_2$ matrix $\mathbf{C}_{\mathbf{Y}\boldsymbol{\Theta}}$ still given by (12). In this work, we always take the weight function $\widetilde{q}(\boldsymbol{\theta})$ to be the sampling density of the training parameters $\{\boldsymbol{\theta}_j\}_{j=1,\ldots,M}$ for the simplification of the expansion coefficients $\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta},k}$ in (15). This assumption is not too restrictive since the training parameters are specified by the users.

Finally, the formula in (10) combined with the expansion coefficients $\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta},k}$ in (15) and the matrix $\mathbf{C}_{\mathbf{Y}\boldsymbol{\Theta}}$ in (12) forms an RKWHS representation of the conditional density $p(\mathbf{y}|\boldsymbol{\theta})$ for arbitrary $\mathbf{y}$ and $\boldsymbol{\theta}$. Numerically, the training outcome is the matrix $\mathbf{C}_{\mathbf{Y}\boldsymbol{\Theta}}$ in (12), and then, the conditional density $\widehat{p}(\mathbf{y}|\boldsymbol{\theta})$ can be represented by (10) with coefficients (15) using the basis functions $\{\psi_k(\mathbf{y})\}_{k=1}^{K_1}$ and $\{\varphi_l(\boldsymbol{\theta})\}_{l=1}^{K_2}$. From above, one can see that two important questions naturally arise as a consequence of the usage of RKWHS representation: first, whether the representation $\widehat{p}(\mathbf{y}|\boldsymbol{\theta})$ in (10) is valid in estimating the conditional density $p(\mathbf{y}|\boldsymbol{\theta})$; second, how to construct the orthonormal basis functions $\psi_k(\mathbf{y}) \in L^2(\mathcal{N}, q)$ and $\varphi_l(\boldsymbol{\theta}) \in L^2(\mathcal{M}, \widetilde{q})$. We will address these two important questions in the next two sections.

3. Error Estimation

In this section, we focus on the error estimation of the expansion coefficient $\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta},k}$ and, later, the conditional density $\widehat{p}(\mathbf{y}|\boldsymbol{\theta}_j)$ at the training parameter $\boldsymbol{\theta}_j$. The notation $\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k}$ is defined as the expansion coefficient $\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta},k}$ in (15), evaluated at the training parameter $\boldsymbol{\theta}_j$. Let the total number of basis functions in parameter space, K_2, be equal to the total number of training parameters, M, that is, $K_2 = M$. Denoting $\boldsymbol{\Phi} = [\vec{\varphi}_1, \ldots, \vec{\varphi}_M] \in \mathbb{R}^{M \times M}$, where the jth component of $\vec{\varphi}_l$ approximates the basis function evaluated at the training data $\varphi_l(\boldsymbol{\theta}_j)$, we can write the last equality in (14) in a compact form as $M^{-1}\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi} = \boldsymbol{I}_M$. This also means that, $M^{-1}\boldsymbol{\Phi}\boldsymbol{\Phi}^{\top} = \boldsymbol{I}_M$, the components of which are,

$$\frac{1}{M} \sum_{l=1}^{M} \varphi_l(\boldsymbol{\theta}_s) \varphi_l(\boldsymbol{\theta}_j) = \delta_{sj}. \tag{16}$$

For the training parameter $\boldsymbol{\theta}_j$, we can simplify the expansion coefficient $\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k}$ by substituting Equation (12) into Equation (15),

$$\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k} = \sum_{l=1}^{M} [\mathbf{C}_{\mathbf{Y}\boldsymbol{\Theta}}]_{kl} \, \varphi_l(\boldsymbol{\theta}_j) \approx \sum_{l=1}^{M} \left[\frac{1}{MN} \sum_{s=1}^{M} \sum_{i=1}^{N} \psi_k(\mathbf{y}_{i,s}) \varphi_l(\boldsymbol{\theta}_s) \right] \varphi_l(\boldsymbol{\theta}_j) = \frac{1}{N} \sum_{i=1}^{N} \psi_k(\mathbf{y}_{i,j}), \tag{17}$$

where the last equality follows from (16).

3.1. Error Estimation Using Arbitrary Bases

We first study the error estimation for the expansion coefficient $\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k}$. For each training parameter $\boldsymbol{\theta}_j$, the conditional density function $p(\mathbf{y}|\boldsymbol{\theta}_j) \in \mathcal{H}_{q^{-1}}(\mathcal{N})$ can be analytically represented in the form,

$$p(\mathbf{y}|\boldsymbol{\theta}_j) = \sum_{k=1}^{\infty} c_{\mathbf{Y}|\boldsymbol{\theta}_j,k} \psi_k(\mathbf{y}) q(\mathbf{y}), \tag{18}$$

due to the completeness of $L^2(\mathcal{N}, q)$. Here, the analytic expansion coefficient $c_{\mathbf{Y}|\boldsymbol{\theta}_j,k}$ is given by,

$$c_{\mathbf{Y}|\boldsymbol{\theta}_j,k} = \left\langle p\left(\cdot|\boldsymbol{\theta}_j\right), \psi_k \right\rangle. \tag{19}$$

Note that the estimator $\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k}$ in (17) is a Monte Carlo approximation of the expansion coefficient $c_{\mathbf{Y}|\boldsymbol{\theta}_j,k}$ in (19), i.e.,

$$c_{\mathbf{Y}|\boldsymbol{\theta}_j,k} = \left\langle p\left(\cdot|\boldsymbol{\theta}_j\right), \psi_k \right\rangle = \mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}_j}[\psi_k(\mathbf{Y})] \approx \frac{1}{N} \sum_{i=1}^{N} \psi_k\left(\mathbf{y}_{i,j}\right), \tag{20}$$

where the last equality follows from the training dataset $\{\mathbf{y}_{i,j}\}_{i=1,\ldots,N}$, which admits a conditional density $p\left(\mathbf{y}|\boldsymbol{\theta}_j\right)$. Note also that in the following theorems and propositions, the condition $p(\mathbf{y}|\boldsymbol{\theta}_j) \in \mathcal{H}_{q^{-1}}(\mathcal{N})$ is required. In Section 5.2 and Appendix B, we will provide an example to discuss this condition in detail. Next, we provide the unbiasedness and consistency of the estimator $\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k}$.

Proposition 1. *Let $\{\mathbf{y}_{i,j}\}_{i=1,\ldots,N}$ be i.i.d. samples of $\mathbf{Y}|\boldsymbol{\theta}_j$ with density $p(\mathbf{y}|\boldsymbol{\theta}_j)$. Let $p(\mathbf{y}|\boldsymbol{\theta}_j) \in \mathcal{H}_{q^{-1}}(\mathcal{N})$ and $\{\psi_k(\mathbf{y})\}$ form a complete orthonormal basis of $L^2(\mathcal{N}, q)$. Assume that $Var_{\mathbf{Y}|\boldsymbol{\theta}_j}[\psi_k(\mathbf{Y})]$ is finite, then $\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k}$ defined in (17) is an unbiased and consistent estimator for $c_{\mathbf{Y}|\boldsymbol{\theta}_j,k}$ in (19).*

Proof. The estimator $\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k}$ is unbiased,

$$\mathbb{E}\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}_j} \psi_k\left(\mathbf{Y}_{i,j}\right) = c_{\mathbf{Y}|\boldsymbol{\theta}_j,k}. \tag{21}$$

where the expectation is taken with respect to the conditional density $p\left(\mathbf{y}|\boldsymbol{\theta}_j\right)$. If the variance, $Var_{\mathbf{Y}|\boldsymbol{\theta}_j}[\psi_k(\mathbf{Y})]$, is finite, then the variance of $\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k}$ converges to zero as the number of training data $N \to \infty$,

$$\text{Var}\left(\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k}\right) = \frac{1}{N} \text{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}[\psi_k(\mathbf{Y})] \to 0, \quad \text{as } N \to \infty. \tag{22}$$

Then, we can obtain that the estimator $\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k}$ is consistent,

$$Pr\left(\left|\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k} - c_{\mathbf{Y}|\boldsymbol{\theta}_j,k}\right| > \varepsilon\right) \leq \frac{\text{Var}\left(\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k}\right)}{\varepsilon^2} \to 0, \text{ as } N \to \infty, \text{ for } \forall \varepsilon > 0,$$

where Chebyshev's inequality has been used. □

If the estimator of $p(\mathbf{y}|\boldsymbol{\theta}_j)$ is given by the representation with an infinite number of basis functions, $\widetilde{p}(\mathbf{y}|\boldsymbol{\theta}_j) = \sum_{k=1}^{\infty} \widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k} \psi_k(\mathbf{y}) q(\mathbf{y})$, then the estimator $\widetilde{p}(\mathbf{y}|\boldsymbol{\theta}_j)$ is pointwise unbiased for every observation $\mathbf{y}$. However, in the numerical implementation, only a finite number of basis functions can be used in the representation (10). Numerically, the estimator of $p(\mathbf{y}|\boldsymbol{\theta}_j)$ is given by the representation (10) at the training parameter $\boldsymbol{\theta}_j$,

$$\widehat{p}(\mathbf{y}|\boldsymbol{\theta}_j) = \sum_{k=1}^{K_1} \widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k} \psi_k(\mathbf{y}) q(\mathbf{y}).$$

Then, the pointwise error of the estimator, $\widehat{e}(\mathbf{y}|\boldsymbol{\theta}_j)$, can be defined as:

$$
\begin{aligned}
\widehat{e}(\mathbf{y}|\boldsymbol{\theta}_j) &\equiv p(\mathbf{y}|\boldsymbol{\theta}_j) - \widehat{p}(\mathbf{y}|\boldsymbol{\theta}_j) \\
&= \sum_{k=K_1+1}^{\infty} c_{\mathbf{Y}|\boldsymbol{\theta}_j,k} \psi_k(\mathbf{y}) q(\mathbf{y}) + \sum_{k=1}^{K_1} \left(c_{\mathbf{Y}|\boldsymbol{\theta}_j,k} - \widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k} \right) \psi_k(\mathbf{y}) q(\mathbf{y}).
\end{aligned}
\tag{23}
$$

It can be seen that the estimator $\widehat{p}(\mathbf{y}|\boldsymbol{\theta}_j)$ is no longer unbiased or consistent due to the first error term in (23) induced by modes $k > K_1$. Next, we estimate the expectation and the variance of an L^2-norm error of $\widehat{p}(\mathbf{y}|\boldsymbol{\theta}_j)$ for all training parameters $\boldsymbol{\theta}_j$.

Theorem 1. *Let the condition in Proposition 1 be satisfied for all $\{\boldsymbol{\theta}_j\}_{j=1,\dots,M}$, and $Var_{\mathbf{Y}|\boldsymbol{\theta}_j} [\psi_k(\mathbf{Y})]$ be finite for all $k \in \mathbb{N}^+$. Define the L^2-norm error,*

$$
\widehat{e}_{L^2} = \left(\sum_{j=1}^{M} \int_{\mathcal{N}} |\widehat{e}(\mathbf{y}|\boldsymbol{\theta}_j)|^2 q^{-1}(\mathbf{y}) dV \right)^{1/2},
\tag{24}
$$

where $\widehat{e}(\mathbf{y}|\boldsymbol{\theta}_j)$ is the pointwise error in (23), and dV is the volume form inherited by the manifold $\mathcal{N}$ from the ambient space $\mathbb{R}^n$ [18,20]. Then,

$$
\mathbb{E}\left[\widehat{e}_{L^2}\right] \leq \left(\sum_{j=1}^{M} \sum_{k=K_1+1}^{\infty} \left[c_{\mathbf{Y}|\boldsymbol{\theta}_j,k}\right]^2 + \frac{1}{N} \sum_{j=1}^{M} \sum_{k=1}^{K_1} Var_{\mathbf{Y}|\boldsymbol{\theta}_j} [\psi_k(\mathbf{Y})] \right)^{\frac{1}{2}},
\tag{25}
$$

$$
Var\left[\widehat{e}_{L^2}\right] \leq \sum_{j=1}^{M} \sum_{k=K_1+1}^{\infty} \left[c_{\mathbf{Y}|\boldsymbol{\theta}_j,k}\right]^2 + \frac{1}{N} \sum_{j=1}^{M} \sum_{k=1}^{K_1} Var_{\mathbf{Y}|\boldsymbol{\theta}_j} [\psi_k(\mathbf{Y})],
\tag{26}
$$

where $\mathbb{E}$ and Var are defined with respect to the joint distribution of $p(\mathbf{y}|\boldsymbol{\theta}_j)$ for all $\{\boldsymbol{\theta}_j\}_{j=1,\dots,M}$. Moreover, $\mathbb{E}\left[\widehat{e}_{L^2}\right]$ and $Var[\widehat{e}_{L^2}]$ converge to zero as $K_1 \to \infty$ and then $N \to \infty$, where the limiting operations of K_1 and N are not commutative.

Proof. The expectation of $\widehat{e}_{L^2}$ can be estimated as,

$$
\begin{aligned}
(\mathbb{E}\left[\widehat{e}_{L^2}\right])^2 &\leq \mathbb{E}\left[\sum_{j=1}^{M} \int_{\mathcal{N}} |\widehat{e}(\mathbf{y}|\boldsymbol{\theta}_j)|^2 q^{-1}(\mathbf{y}) dV \right] \\
&= \mathbb{E}\left[\sum_{j=1}^{M} \int_{\mathcal{N}} \left(\sum_{k=K_1+1}^{\infty} c_{\mathbf{Y}|\boldsymbol{\theta}_j,k} \psi_k(\mathbf{y}) + \sum_{k=1}^{K_1} \left(c_{\mathbf{Y}|\boldsymbol{\theta}_j,k} - \widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k} \right) \psi_k(\mathbf{y}) \right)^2 q(\mathbf{y}) dV \right],
\end{aligned}
\tag{27}
$$

where the first inequality follows from Jensen's inequality. Here, the randomness comes from the estimators $\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k}$. Due to the orthonormality of basis functions, $\psi_k \in L^2(\mathcal{N}, q)$, the error estimation in (27) can be simplified as,

$$
\begin{aligned}
(\mathbb{E}\left[\widehat{e}_{L^2}\right])^2 &\leq \sum_{j=1}^{M} \sum_{k=K_1+1}^{\infty} \left[c_{\mathbf{Y}|\boldsymbol{\theta}_j,k}\right]^2 + \sum_{j=1}^{M} \sum_{k=1}^{K_1} \mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}_j} \left[\left(c_{\mathbf{Y}|\boldsymbol{\theta}_j,k} - \widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k} \right)^2 \right], \\
&= \sum_{j=1}^{M} \sum_{k=K_1+1}^{\infty} \left[c_{\mathbf{Y}|\boldsymbol{\theta}_j,k}\right]^2 + \frac{1}{N} \sum_{j=1}^{M} \sum_{k=1}^{K_1} Var_{\mathbf{Y}|\boldsymbol{\theta}_j} [\psi_k(\mathbf{Y})],
\end{aligned}
\tag{28}
$$

where the inequality follows from the linearity of expectation, and the equality follows from $\mathbb{E}\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k} = c_{\mathbf{Y}|\boldsymbol{\theta}_j,k}$ in (21) and $Var\left(\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k}\right) = \frac{1}{N} Var_{\mathbf{Y}|\boldsymbol{\theta}_j} [\psi_k(\mathbf{Y})]$ in (22). In error estimation (28), the first term is

deterministic, and the second term is random. We have so far proven that the expectation $\mathbb{E}\left[\widehat{e}_{L^2}\right]$ is bounded by (25). Similarly, we can prove that the variance $\mathrm{Var}[\widehat{e}_{L^2}]$ is bounded by (26).

Next, we prove that the expectation $\mathbb{E}\left[\widehat{e}_{L^2}\right]$ converges to zero as $K_1 \to \infty$ and then $N \to \infty$. Parseval's theorem states that:

$$\sum_{k=1}^{\infty} \left[c_{\mathbf{Y}|\boldsymbol{\theta}_j,k}\right]^2 = \int_{\mathcal{N}} p(\mathbf{y}|\boldsymbol{\theta}_j)^2 q^{-1}(\mathbf{y})\, dV < +\infty, \quad \text{for all } \boldsymbol{\theta}_j, \tag{29}$$

where the inequality follows from $p(\mathbf{y}|\boldsymbol{\theta}_j) \in \mathcal{H}_{q^{-1}}(\mathcal{N}) \subset L^2\left(\mathcal{N}, q^{-1}\right)$ for all $\boldsymbol{\theta}_j$. For $\forall \varepsilon > 0$, there exists an integer $\widetilde{K}_1\left(\boldsymbol{\theta}_j\right)$ for $\boldsymbol{\theta}_j$ such that:

$$\sum_{k=\widetilde{K}_1(\boldsymbol{\theta}_j)}^{\infty} \left[c_{\mathbf{Y}|\boldsymbol{\theta}_j,k}\right]^2 < \frac{\varepsilon}{2M}. \tag{30}$$

Let:

$$K_1 = \max\left\{\widetilde{K}_1\left(\boldsymbol{\theta}_1\right), \ldots, \widetilde{K}_1\left(\boldsymbol{\theta}_M\right)\right\}, \tag{31}$$

then the first term in (28) can be bounded by: $\varepsilon/2$,

$$\sum_{j=1}^{M} \sum_{k=K_1+1}^{\infty} \left[c_{\mathbf{Y}|\boldsymbol{\theta}_j,k}\right]^2 < \frac{\varepsilon}{2}. \tag{32}$$

Since the variance $\mathrm{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}\left[\psi_k(\mathbf{Y})\right]$ is assumed to be finite for all k and j, there exists a constant $D > 0$ such that $\mathrm{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}\left[\psi_k(\mathbf{Y})\right]$ can be bounded above by this constant D,

$$\mathrm{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}\left[\psi_k(\mathbf{Y})\right] \le D, \quad \text{for all } k = 1, \ldots, K_1 \text{ and } j = 1, \ldots, M. \tag{33}$$

Then, for $\forall \varepsilon > 0$, there exists a sufficiently large number of training data:

$$N_{\min} = \frac{2MK_1 D}{\varepsilon}. \tag{34}$$

such that whenever $N > N_{\min}$, then:

$$\frac{1}{N} \sum_{j=1}^{M} \sum_{k=1}^{K_1} \mathrm{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}\left[\psi_k(\mathbf{Y})\right] < \frac{\varepsilon}{2}. \tag{35}$$

Since $\varepsilon > 0$ is arbitrary, by substituting Equation (32) and Equation (35) into the error estimation (28), we obtain that $\mathbb{E}\left[\widehat{e}_{L^2}\right]$ converges to zero as $K_1 \to \infty$ and then $N \to \infty$. Note that, we first take $K_1 \to \infty$ to ensure the first error term in (28) vanishes and then take $N \to \infty$ to ensure the second error term in (28) vanishes. Thus, the limiting operations of K_1 and N are not commutative. Similarly, we can prove that the variance $\mathrm{Var}[\widehat{e}_{L^2}]$ converges to zero as $K_1 \to \infty$ and then $N \to \infty$. □

Theorem 1 provides the intuition for specifying the number of training observation data N to achieve any desired accuracy $\varepsilon > 0$ given fixed M-parameters and sufficiently large K_1. It can be seen from Theorem 1 that numerically, the expectation $\mathbb{E}\left[\widehat{e}_{L^2}\right]$ in (25) and the variance $\mathrm{Var}[\widehat{e}_{L^2}]$ in (26) can be bounded within arbitrarily small ε by choosing sufficiently large K_1 and N. Specifically, there are two error terms in Equations (25) and (26), the first being deterministic, induced by modes $k > K_1$, and the second random, induced by modes $k \le K_1$. For the deterministic term ($k > K_1$), the error can be bounded by $\varepsilon/2$ by choosing sufficiently large K_1 satisfying (31). In our implementation, the number of basis functions K_1 is empirically chosen to be large enough in order to make the first error term in Equations (25) and (26) for $k > K_1$ as small as possible.

For the random term ($k \leq K_1$), the error can be bounded by $\varepsilon/2$ by choosing sufficiently large N satisfying $N > N_{\min} = 2MK_1D/\varepsilon$ (Equation (34)). The minimum number of training data, $N_{\min}$, depends on the upper bound of $\text{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}[\psi_k(\mathbf{Y})]$, D. However, the upper bound D may not exist for some problems. This means that for some problems, the assumption for finite $\text{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}[\psi_k(\mathbf{Y})]$ in Theorem 1 may not be satisfied. Even if the upper bound D exists, it is typically not easy to evaluate its value given an arbitrary basis $\psi_k \in L^2(\mathcal{N}, q)$ since one needs to evaluate $\text{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}[\psi_k(\mathbf{Y})]$ for all $k = 1, \ldots, K_1$ and $j = 1, \ldots, M$. Note that Theorem 1 holds true for representing $\widehat{p}(\mathbf{y}|\boldsymbol{\theta}_j)$ with an arbitrary basis $\{\psi_k\} \in L^2(\mathcal{N}, q)$ as long as $p(\mathbf{y}|\boldsymbol{\theta}_j) \in \mathcal{H}_{q^{-1}}(\mathcal{N})$ for all θ_j and $\text{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}[\psi_k(\mathbf{Y})]$ is finite for $k \leq K_1$ and $j = 1, \ldots, M$. Next, we provide several cases in which $\text{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}[\psi_k(\mathbf{Y})]$ is finite for k and j.

Remark 1. *If the weighted Hilbert space $L^2(\mathcal{N}, q)$ is defined on a compact manifold $\mathcal{N}$ and has smooth basis functions ψ_k, then $\text{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}[\psi_k(\mathbf{Y})]$ is finite for a fixed $k \in \mathbb{N}^+$ and $j = 1, \ldots, M$. This assertion follows from the fact that continuous functions on a compact manifold are bounded. The smoothness assumption is not unreasonable in many applications since the orthonormal basis functions are obtained as solutions of an eigenvalue problem of a self-adjoint second-order elliptic differential operator. Note that the bound here is not necessarily a uniform bound of $\psi_k(\mathbf{Y})$ for all $k \in \mathbb{N}^+$ and $j = 1, \ldots, M$. As long as $\text{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}[\psi_k(\mathbf{Y})]$ is finite for $k \leq K_1$ and $j = 1, \ldots, M$, the upper bound D is finite, and then, Theorem 1 holds.*

Remark 2. *If the manifold $\mathcal{N}$ is a hyperrectangle in $\mathbb{R}^n$ and the weight q is a uniform distribution on $\mathcal{N}$, then $\text{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}[\psi_k(\mathbf{Y})]$ is finite for a fixed $k \in \mathbb{N}^+$ and $j = 1, \ldots, M$. This assertion is an immediate consequence of Remark 1.*

In Theorem 1, $N_{\min}$ depends on the upper bound of $\text{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}[\psi_k(\mathbf{Y})]$, D, as shown in (34). In the following, we will specify a Hilbert space, referred to as a data-driven Hilbert space, so that $N_{\min}$ is independent of D and is only dependent of M, K_1, and ε. As a consequence, we can easily determine how many training data N for bounding the second error term in Equations (25) and (26).

3.2. Error Estimation Using a Data-Driven Hilbert Space

We now turn to the discussion of a specific data-driven Hilbert space $L^2(\mathcal{N}, \overline{q})$ with orthonormal basis functions $\overline{\psi}_k$. Our goal is to specify the weight function $\overline{q}$ such that the minimum number of training data, $\overline{N}_{\min}$, only depends on M, K_1, and ε. Here, the overline $\overline{\cdot}$ corresponds to the specific data-driven Hilbert space. The second error term in (28) can be further estimated as,

$$\frac{1}{N}\sum_{j=1}^{M}\sum_{k=1}^{K_1}\text{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}\left[\overline{\psi}_k(\mathbf{Y})\right] \leq \frac{1}{N}\sum_{k=1}^{K_1}\sum_{j=1}^{M}\mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}_j}\left[\overline{\psi}_k^2(\mathbf{Y})\right] = \frac{M}{N}\sum_{k=1}^{K_1}\int_{\mathcal{N}}\overline{\psi}_k^2(\mathbf{y})\left(\frac{1}{M}\sum_{j=1}^{M}p(\mathbf{y}|\boldsymbol{\theta}_j)\right)dV, \quad (36)$$

where the basis functions are substituted with the specific $\overline{\psi}_k$. Notice that $\overline{\psi}_k(\mathbf{y})$ are orthonormal basis functions with respect to the weight $\overline{q}$ in $L^2(\mathcal{N}, \overline{q})$. One specific choice of the weight function $\overline{q}(\mathbf{y})$ is:

$$\overline{q}(\mathbf{y}) = \frac{1}{M}\sum_{j=1}^{M}p(\mathbf{y}|\boldsymbol{\theta}_j), \quad (37)$$

where $\overline{q}(\mathbf{y})$ has been normalized, i.e.,

$$\int_{\mathcal{N}}\overline{q}(\mathbf{y})\,dV = \int_{\mathcal{N}}\frac{1}{M}\sum_{j=1}^{M}p(\mathbf{y}|\boldsymbol{\theta}_j)dV = 1. \quad (38)$$

For the data-driven Hilbert space, we always use a normalized weight function $\bar{q}(\mathbf{y})$. Note that the weight function $\bar{q}(\mathbf{y})$ in (37) is a discretization of the marginal density function of $\mathbf{Y}$ with $\boldsymbol{\Theta}$ marginalized out,

$$\bar{q}(\mathbf{y}) = \frac{1}{M} \sum_{j=1}^{M} p(\mathbf{y}|\boldsymbol{\theta}_j) \approx \int_{\mathcal{M}} p(\mathbf{y}|\boldsymbol{\theta}) \widetilde{q}(\boldsymbol{\theta}) \, d\boldsymbol{\theta} = \int_{\mathcal{M}} p(\mathbf{y}, \boldsymbol{\theta}) d\boldsymbol{\theta}, \tag{39}$$

where $p(\mathbf{y}, \boldsymbol{\theta})$ denotes the joint density of $(\mathbf{Y}, \boldsymbol{\Theta})$. Essentially, the weight function $\bar{q}(\mathbf{y})$ in (37) is the sampling density of all the training data $\{\mathbf{y}_{i,j}\}_{j=1,\dots,M}^{i=1,\dots,N}$, which motivates us to refer to $L^2(\mathcal{N}, \bar{q})$ as a data-driven Hilbert space.

Next, we prove that by specifying the data-driven basis functions $\bar{\psi}_k \in L^2(\mathcal{N}, \bar{q})$, the variance $\mathrm{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}[\bar{\psi}_k(\mathbf{Y})]$ is finite for all $k \in \mathbb{N}^+$ and $j = 1, \dots, M$. Subsequently, we can obtain the minimum number of training data, $\overline{N}_{\min}$, to only depend on M, K_1, and ε, such that the expectation $\mathbb{E}[\widehat{e}_{L^2}]$ in (25) and the variance $\mathrm{Var}[\widehat{e}_{L^2}]$ in (26) are bounded above by any $\varepsilon > 0$.

Proposition 2. *Let $\{\mathbf{y}_{i,j}\}_{i=1,\dots,N}$ be i.i.d. samples of $\mathbf{Y}|\boldsymbol{\theta}_j$ with density $p(\mathbf{y}|\boldsymbol{\theta}_j)$. Let $p(\mathbf{y}|\boldsymbol{\theta}_j) \in \mathcal{H}_{q^{-1}}(\mathcal{N})$ for all $\{\boldsymbol{\theta}_j\}_{j=1,\dots,M}$ with weight $\bar{q}$ specified in (37), and let $\{\bar{\psi}_k\}$ be the complete orthonormal basis of $L^2(\mathcal{N}, \bar{q})$. Then, $\mathrm{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}[\bar{\psi}_k(\mathbf{Y})]$ is finite for all $k \in \mathbb{N}^+$ and $j = 1, \dots, M$.*

Proof. Notice that for all $k \in \mathbb{N}^+$, we have:

$$\begin{aligned}
\frac{1}{M} \sum_{j=1}^{M} \mathrm{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}[\bar{\psi}_k(\mathbf{Y})] &\leq \frac{1}{M} \sum_{j=1}^{M} \mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}_j}\left[\bar{\psi}_k^2(\mathbf{Y})\right] = \int_{\mathcal{N}} \bar{\psi}_k^2(\mathbf{y}) \frac{1}{M} \sum_{j=1}^{M} p(\mathbf{y}|\boldsymbol{\theta}_j) dV \\
&= \int_{\mathcal{N}} \bar{\psi}_k^2(\mathbf{y}) \bar{q}(\mathbf{y}) \, dV = 1,
\end{aligned} \tag{40}$$

where the last equality follows directly from the orthonormality of basis functions $\bar{\psi}_k(\mathbf{y}) \in L^2(\mathcal{N}, \bar{q})$. From Equation (40), we can obtain that for all $k \in \mathbb{N}^+$ and $j = 1, \dots, M$, the variance $\mathrm{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}[\bar{\psi}_k(\mathbf{Y})]$ is finite. $\square$

Theorem 2. *Given the same hypothesis as in Proposition 2, then:*

$$\mathbb{E}[\widehat{e}_{L^2}] \leq \left(\sum_{j=1}^{M} \sum_{k=K_1+1}^{\infty} \left[c_{\mathbf{Y}|\boldsymbol{\theta}_j,k}\right]^2 + \frac{MK_1}{N}\right)^{\frac{1}{2}}, \tag{41}$$

$$\mathrm{Var}[\widehat{e}_{L^2}] \leq \sum_{j=1}^{M} \sum_{k=K_1+1}^{\infty} \left[c_{\mathbf{Y}|\boldsymbol{\theta}_j,k}\right]^2 + \frac{MK_1}{N}. \tag{42}$$

where $\widehat{e}_{L^2}$ is defined by (24) and $c_{\mathbf{Y}|\boldsymbol{\theta}_j,k}$ is given by (19). Moreover, $\mathbb{E}[\widehat{e}_{L^2}]$ and $\mathrm{Var}[\widehat{e}_{L^2}]$ converge to zero as $K_1 \to \infty$ and then $N \to \infty$, where the limiting operations of K_1 and N are not commutative.

Proof. According to Proposition 2, we have that the variance $\mathrm{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}[\bar{\psi}_k(\mathbf{Y})]$ is finite for all $k \in \mathbb{N}^+$ and $j = 1, \dots, M$. According to Proposition 1, since $\mathrm{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}[\bar{\psi}_k(\mathbf{Y})]$ is finite, we have that the estimator $\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta}_j,k}$ is both unbiased and consistent for $c_{\mathbf{Y}|\boldsymbol{\theta}_j,k}$. All conditions in Theorem 1 are satisfied, so that we can obtain the error estimation of the expectation $\mathbb{E}[\widehat{e}_{L^2}]$ in (25) and the error estimation of the variance $\mathrm{Var}[\widehat{e}_{L^2}]$ in (26). Moreover, the second error term in $\mathbb{E}[\widehat{e}_{L^2}]$ (25) and $\mathrm{Var}[\widehat{e}_{L^2}]$ (26) can be both bounded by Equation (40), so that we can obtain our error estimations (41) and (42).

Choose K_1 as in (31) such that the first term in (41) and (42) is bounded by $\varepsilon/2$. The second term MK_1/N in (41) and (42) can be bounded by an arbitrarily small $\varepsilon/2$ if the number of training data N satisfies:

$$N > \overline{N}_{\min} \equiv 2MK_1/\varepsilon. \tag{43}$$

Then, both the expectation $\mathbb{E}\left[\hat{e}_{L^2}\right]$ and the variance $\text{Var}[\hat{e}_{L^2}]$ can be bounded by ε. Since $\varepsilon > 0$ is arbitrary, the proof is complete. $\square$

Recall that by applying arbitrary basis functions to represent $\hat{p}\left(\mathbf{y}|\boldsymbol{\theta}\right)$ in (10), it is typically not easy to evaluate the upper bound D in (33), which implies that it is not easy to determine how many observation data, $N_{\min}$ (Equation (34)), should be used for training. However, by applying the data-driven basis functions $\overline{\psi}_k$ to represent $\hat{p}\left(\mathbf{y}|\boldsymbol{\theta}\right)$ in (10), the minimum number of training data, $\overline{N}_{\min}$ (Equation (43)), becomes independent of D, and is only dependent of M, K_1, and ε, as can be seen from Theorem 2. To let the error induced by modes $k \leq K_1$ be smaller than a desired $\varepsilon/2$, we can easily determine how many observation data, $\overline{N}_{\min}$ (Equation (43)), should be used for training. In this sense, the specific data-driven Hilbert space $L^2\left(\mathcal{N},\overline{q}\right)$ with the corresponding basis functions $\overline{\psi}_k$ is a good choice for representing (10).

We have so far theoretically verified the validity of the representation (10) in estimating the conditional density $p(\mathbf{y}|\boldsymbol{\theta}_j)$ (Theorem 1). In particular, using the data-driven basis $\overline{\psi}_k \in L^2\left(\mathcal{N},\overline{q}\right)$, we can easily control the error of conditional density estimation by specifying the number of training data N (Theorem 2). To summarize, the training procedures can be outlined as follows:

(1-A) Generate the training dataset, including training parameters $\left\{\boldsymbol{\theta}_j\right\}_{j=1,\ldots,M}$ and observations $\left\{\mathbf{y}_{i,j}\right\}_{j=1,\ldots,M}^{i=1,\ldots,N}$. The length of training data N is empirically determined based on the criteria (34) or (43).

(1-B) Construct the basis functions for parameter $\boldsymbol{\theta}$ space and for observation $\mathbf{y}$ space by using the training dataset. For $\mathbf{y}$ space, we need to empirically choose the number of basis functions K_1 to let the error induced by modes $k > K_1$ be as small as possible. In particular, for the data-driven Hilbert space, we will provide a detailed discussion on how to estimate the data-driven basis functions of $L^2(\mathcal{N},\overline{q})$ with the sampling density $\overline{q}$ from the training data in the following Section 4. Note that this basis estimation will introduce additional errors beyond the results in this section, which assumed the data-driven basis functions to be given.

(1-C) Train the matrix $\mathbf{C}_{\mathbf{Y\Theta}}$ in (12) and then estimate the conditional density $\hat{p}\left(\mathbf{y}|\boldsymbol{\theta}\right)$ by using the nonparametric RKWHS representation (10) with the expansion coefficients $\hat{c}_{\mathbf{Y}|\boldsymbol{\theta},k}$ (15).

(1-D) Finally, for new observations $\mathbf{y}^{\dagger} = \{\mathbf{y}_1^{\dagger},\ldots,\mathbf{y}_T^{\dagger}\}$, define the likelihood function as a product of the conditional densities of new observations $\mathbf{y}^{\dagger}$ given any $\boldsymbol{\theta}$,

$$p(\mathbf{y}^{\dagger}|\boldsymbol{\theta}) \equiv \prod_{t=1}^{T} \hat{p}(\mathbf{y}_t^{\dagger}|\boldsymbol{\theta}). \tag{44}$$

Next, we address the second important question for the RKWHS representation (Procedure (1-B)): how to construct basis functions for $\boldsymbol{\theta}$ and $\mathbf{y}$. Especially, we focus on how to construct the data-driven basis functions for $\mathbf{y}$.

4. Basis Functions

This section will be organized as follows. In Section 4.1, we discuss how to employ analytical basis functions for parameter $\boldsymbol{\theta}$ and for observation $\mathbf{y}$ as in the usual polynomial chaos expansion. In Section 4.2, we discuss how to construct the data-driven basis functions $\overline{\psi}_k \in L^2\left(\mathcal{N},\overline{q}\right)$ with $\mathcal{N}$ being the manifold of the training dataset $\left\{\mathbf{y}_{i,j}\right\}_{j=1,\ldots,M}^{i=1,\ldots,N}$ and the weight $\overline{q}$ by (37) being the sampling density of $\left\{\mathbf{y}_{i,j}\right\}_{j=1,\ldots,M}^{i=1,\ldots,N}$.

4.1. Analytic Basis Functions

If no prior information about the parameter space other than its domain is known, we can assume that the training parameters are uniformly distributed on the parameter $\boldsymbol{\theta}$ space. In particular, we choose M number of well-sampled training parameters $\{\boldsymbol{\theta}_j\}_{j=1,\ldots,M} = \{\theta_j^1,\ldots,\theta_j^m\}_{j=1,\ldots,M}$ in an m-dimensional box $\mathcal{M} \subset \mathbb{R}^m$,

$$\mathcal{M} \equiv [\theta_{\min}^1, \theta_{\max}^1] \times \cdots \times [\theta_{\min}^m, \theta_{\max}^m], \tag{45}$$

where $\times$ denotes a Cartesian product and the two parameters $\theta_{\min}^s$ and $\theta_{\max}^s$ are the minimum and maximum values of the uniform distribution for the sth coordinate of $\boldsymbol{\theta}$ space. Here, the well-sampled uniform distribution corresponds to a regular grid, which is a tessellation of m-dimensional Euclidean space $\mathbb{R}^m$ by congruent parallelotopes. Two parameters $\theta_{\min}^s$ and $\theta_{\max}^s$ are determined by:

$$
\begin{aligned}
\theta_{\min}^s &= \min_{j=1,\ldots,M}\left\{\theta_j^s\right\} - \gamma\big(\max_{j=1,\ldots,M}\left\{\theta_j^s\right\} - \min_{j=1,\ldots,M}\left\{\theta_j^s\right\}\big), \\
\theta_{\max}^s &= \max_{j=1,\ldots,M}\left\{\theta_j^s\right\} + \gamma\big(\max_{j=1,\ldots,M}\left\{\theta_j^s\right\} - \min_{j=1,\ldots,M}\left\{\theta_j^s\right\}\big).
\end{aligned}
\tag{46}
$$

For M regularly-spaced grid points θ_j^s, we set $\gamma = .5(M^s - 1)^{-1}$ in all of our numerical examples below, where M^s is the number of training parameters in the sth coordinate. For example, see Figure 1 for the 2D well-sampled uniformly-distributed data $\{(5,5),(6,5),\ldots,(12,12)\}$ (blue circles). In this case, the two-dimensional box $\mathcal{M}$ is $[4.5, 12.5]^2$ (red square).
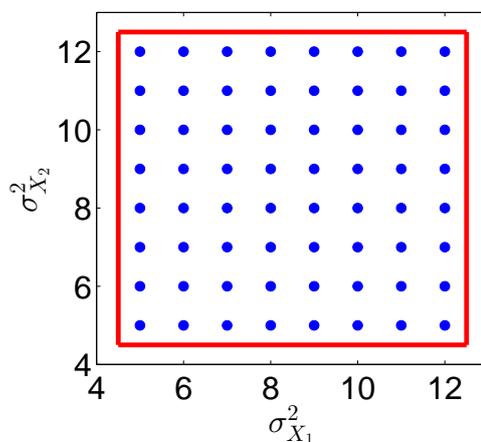

Figure 1. (Color online) An example of well-sampled 2D uniformly-distributed data points (blue circles). The boundary of the uniform distribution is depicted with a red square. Furthermore, these well-sampled data points correspond to the training parameters in Example I in Section 5. In this example, the well-sampled uniformly-distributed training parameters are $(\sigma_{X_1}^2, \sigma_{X_2}^2) \in \{(i,j)\}_{i=5,\ldots,12}^{j=5,\ldots,12}$ (blue circles). The equal spacing distances of both coordinates are one. The two-dimensional box $\mathcal{M}$ is $[4.5, 12.5]^2$ (red square).

On this simple geometry, we will choose φ_k to be the tensor product of the basis functions on each coordinate. Notice that we have taken the weight function $\widetilde{q}$ to be the sampling density of the training parameters in order to simplify the expansion coefficient $\widehat{c}_{\mathbf{Y}|\boldsymbol{\theta},k}$ in (15). In this case, the weight $\widetilde{q}$ is a uniform distribution on $\mathcal{M}$. Then, for the sth coordinate of the parameter, θ^s, the weight function $\widehat{q}^s(\theta^s)$ is a uniform distribution on the interval $[\theta_{\min}^s, \theta_{\max}^s]$, and one can choose the following cosine basis functions,

$$\Phi_{k^s}(\theta^s) = \begin{cases} 1, & \text{if } k^s = 0, \\ \sqrt{2}\cos\left(k^s \pi \frac{\theta^s - \theta_{\min}^s}{\theta_{\max}^s - \theta_{\min}^s}\right), & \text{else,} \end{cases} \tag{47}$$

where $\Phi_{k^s}(\theta^s)$ form a complete orthonormal basis of $L^2\left([\theta^s_{\min},\theta^s_{\max}],\widetilde{q}^s\right)$. This choice of basis functions corresponds to exactly the data-driven basis functions produced by the diffusion maps algorithm on the uniformly-distributed dataset on a compact interval, which will be discussed in Section 4.2. Although other choices such as the Legendre polynomials can be used, this choice will lead to a larger value of constant D in (34) that controls the minimum number of training data for accurate estimation.

Subsequently, we set $L^2\left(\mathcal{M},\widetilde{q}\right) = \bigotimes_{s=1}^m L^2\left([\theta^s_{\min},\theta^s_{\max}],\widetilde{q}^s\right)$, where $\otimes$ denotes the Hilbert tensor product, and $\widetilde{q}\left(\boldsymbol{\theta}\right) = \bigotimes_{s=1}^m \widetilde{q}^s(\theta^s)$ is the uniform distribution on the m-dimensional box $\mathcal{M}$. Correspondingly, the basis functions $\varphi_k\left(\boldsymbol{\theta}\right)$ are a tensor product of $\Phi_{k^s}(\theta^s)$ for $s = 1,\dots,m$,

$$\varphi_k\left(\boldsymbol{\theta}\right) = \bigotimes_{s=1}^m \Phi_{k^s}(\theta^s) = \Phi_{k^1}(\theta^1) \otimes \dots \otimes \Phi_{k^m}(\theta^m), \tag{48}$$

where $k = \left(k^1,\dots,k^m\right)$ and $\boldsymbol{\theta} = \left(\theta^1,\dots,\theta^m\right)$. Based on the property of the tensor product of Hilbert spaces, $\{\varphi_k\left(\boldsymbol{\theta}\right)\}$ forms a complete orthonormal basis of $L^2\left(\mathcal{M},\widetilde{q}\right)$.

We now turn to the discussion of how to construct analytic basis functions for $\mathbf{y}$. The approach is similar to the one for parameter $\boldsymbol{\theta}$, except that the domain of the data is specified empirically and the weight function is chosen to correspond to some well-known analytical basis functions, independent of the sampling distribution of the data $\mathbf{y}$. That is, we assume the geometry of the data has the following tensor structure, $\mathcal{N} = \mathcal{N}^1 \times \dots \times \mathcal{N}^n$, where $\mathcal{N}^s$ will be specified empirically based on the ambient space coordinate of $\mathbf{y}$. Let y^s be the sth ambient component of $\mathbf{y}$; we can choose a weighted Hilbert space $L^2\left(\mathcal{N}^s, q^s(y^s;\alpha^s)\right)$ with the weight q^s depending on the parameters α^s and being normalized to satisfy $\int_{\mathbb{R}} q^s(y^s;\alpha^s)dy^s = 1$. For each coordinate, let $\Psi_{k^s}(y^s;\alpha^s)$ be the corresponding orthonormal basis functions, which possess analytic expressions. Subsequently, we can obtain a set of complete orthonormal basis functions $\psi_k \in L^2\left(\mathcal{N},q\right)$ for $\mathbf{y}$ by taking the tensor product of these Ψ_{k^s} as in (48).

For example, if the weight q^s is uniform, $\mathcal{N}^s \subset \mathbb{R}$ is simply a one-dimensional interval. In this case, we can choose the cosine basis functions Ψ_{k^s} for $\mathbf{y}$ as in (47) such that the parameters α^s correspond to the boundaries of the domain $\mathcal{N}^s$, which can be estimated as in (46). In our numerical experiments below, we will set $\gamma = 0.1$. Another choice is to set the weight $q^s(y^s;\alpha^s)$ to be Gaussian. In this case, the domain is assumed to be the real line, $\mathcal{N}^s = \mathbb{R}$. For this choice, the corresponding orthonormal basis functions Ψ_{k^s} are Hermite polynomials, and the parameters α^s, corresponding to the mean and variance of the Gaussian distribution, can be empirically estimated from the training data.

In the remainder of this paper, we will always use the cosine basis functions for $\boldsymbol{\theta}$. The application of (10) using cosine basis functions for $\mathbf{y}$ is referred to as the cosine representation. The application of (10) using Hermite basis functions for $\mathbf{y}$ is referred to as the Hermite representation.

4.2. Data-Driven Basis Functions

In this section, we discuss how to construct a set of data-driven basis functions $\overline{\psi}_k \in L^2\left(\mathcal{N},\overline{q}\right)$ with $\mathcal{N}$ being the manifold of the training dataset $\{\mathbf{y}_{i,j}\}_{j=1,\dots,M}^{i=1,\dots,N}$ and weight $\overline{q}$ in (37) being the sampling density of $\{\mathbf{y}_{i,j}\}$ for all $i = 1,\dots,N$, and $j = 1,\dots,M$. The issues here are that the analytical expression of the sampling density $\overline{q}$ is unknown and the Riemannian metric inherited by the data manifold $\mathcal{N}$ from the ambient space $\mathbb{R}^n$ is also unknown. Fortunately, these issues can be overcome by the diffusion maps algorithm [18–20].

4.2.1. Learning the Data-Driven Basis Functions

Given a dataset $\mathbf{y}_{i,j} \in \mathcal{N} \subseteq \mathbb{R}^n$ with the sampling density $\overline{q}(\mathbf{y})$ (37), defined with respect to the volume form inherited by the manifold $\mathcal{N}$ from the ambient space $\mathbb{R}^n$, one can use the kernel-based diffusion maps algorithm to construct an $MN \times MN$ matrix $\mathbf{L}$ that approximates a weighted Laplacian operator, $\mathcal{L} = \nabla \log\left(\overline{q}\right) \cdot \nabla + \triangle$, that takes functions with Neumann boundary conditions for the compact manifold $\mathcal{N}$ with the boundary if the manifold has a boundary. The eigenvectors $\overline{\boldsymbol{\psi}}_k$ of the matrix $\mathbf{L}$ are discrete approximations of the eigenfunctions $\overline{\psi}_k\left(\mathbf{y}\right)$ of the operator $\mathcal{L}$, which form

an orthonormal basis of the weighted Hilbert space $L^2(\mathcal{N}, \bar{q})$. Connecting to the discussion on the RKWHS in Section 2, the eigenfunctions of $\mathcal{L}^* = -\text{div}(\nabla \log \bar{q}) + \Delta$, that is $\{\Psi_k := \bar{\psi}_k \bar{q}\}$, can be approximated using an integral operator in (7) with the appropriate kernel constructed by the diffusion maps algorithm, up to a diagonal conjugation. Basically, $\mathcal{H}_{\bar{q}^{-1}}(\mathcal{N})$ is the data-driven reproducing kernel Hilbert space defined with the feature map in (9), induced by eigenfunctions of $\mathcal{L}^*$.

Each component of the eigenvector $\overline{\boldsymbol{\psi}}_k \in \mathbb{R}^{MN}$ is a discrete estimate of the eigenfunction $\overline{\psi}_k(\mathbf{y}_{i,j})$, evaluated at the training data point $\mathbf{y}_{i,j}$. The sampling density $\bar{q}$ defined in (37) is estimated using a kernel density estimation method [22]. In contrast to the analytic continuous basis functions in the above Section 4.1, the data-driven basis functions $\overline{\psi}_k \in L^2(\mathcal{N}, \bar{q})$ are represented nonparametrically by the discrete eigenvectors $\overline{\boldsymbol{\psi}}_k \in \mathbb{R}^{MN}$ using the diffusion maps algorithm. The outcome of the training is a discrete estimate of the conditional density, $\widehat{p}(\mathbf{y}_{i,j}|\boldsymbol{\theta})$, which estimates the representation $\widehat{p}(\mathbf{y}|\boldsymbol{\theta})$ (10) on each training data point $\mathbf{y}_{i,j}$.

In our implementation, we use the Variable-Bandwidth Diffusion Maps (VBDM) algorithm introduced in [20], which extends the diffusion maps to non-compact manifolds without a boundary. See the supplementary material of [23] for the MATLAB code of this algorithm. We should point out that this discrete approximation induces errors in the basis function, which are estimated in detail in [24]. These errors are in addition to the error estimations in Section 3.

We note that if the data are uniformly distributed on a one-dimensional bounded interval, then the VBDM solutions are the cosine basis functions, which are eigenfunctions of the Laplacian operator on bounded interval with Neumann boundary conditions. This means that the cosine functions in (47) that are used to represent each component of $\boldsymbol{\theta}$ are analogous to the data-driven basis functions. The difference is that with the parametric choice in (47), one avoids VBDM at the expense of specifying the boundaries of the domain, $[\theta^s_{\min}, \theta^s_{\max}]$. In the remainder of this paper, we refer to an application of (10) with cosine basis functions for $\boldsymbol{\theta}$ and VBDM basis functions for $\mathbf{y}$ as the VBDM representation.

However, a direct application of the VBDM algorithm suffers from the expensive computational cost for large training data. Basically, we need an algorithm that allows us to subsample from the training dataset while preserving the sampling distribution of the full dataset. In Appendix A, we provide a simple box-averaging method to achieve this goal. In the remainder of this paper, we will denote the reduced data obtained via the box-averaging method in Appendix A by $\{\overline{\mathbf{y}}_b\}_{b=1,\dots,B}$, where $B \ll MN$. We refer to them as the box-averaged data points. When the number of training data is too large, we apply the VBDM algorithm on these box-averaged data to obtain the discrete estimate of the eigenfunctions $\overline{\psi}_k(\overline{\mathbf{y}}_b)$.

The second issue arises from the discrete representation of the conditional density in the observation $\mathbf{y}$ space using the VBDM algorithm. Notice that the VBDM representation, $\widehat{p}(\mathbf{y}_{i,j}|\boldsymbol{\theta})$, is only estimated at each training data point $\mathbf{y}_{i,j}$. A natural problem is to extend the representation onto new observations $\mathbf{y}_t \notin \{\mathbf{y}_{i,j}\}_{j=1,\dots,M}^{i=1,\dots,N}$ that are not part of the training dataset (Procedure (1-D)). Next, we address this issue.

4.2.2. Nyström Extension

We now discuss an extension method to evaluate basis functions $\overline{\psi}_k$ on a new data point that does not belong to the training dataset. Given such an extension method, we can proceed with Procedure (1-D) by evaluating $\overline{\psi}_k(\mathbf{y}_t)$ on new observations $\mathbf{y}_t \notin \{\mathbf{y}_{i,j}\}_{j=1,\dots,M}^{i=1,\dots,N}$, which in turn give $\widehat{p}(\mathbf{y}_t|\boldsymbol{\theta})$. Second, this extension is also needed in the training Procedure (1-C) when MN is large. More specifically, for training the matrix $\mathbf{C}_{\mathbf{Y}\boldsymbol{\Theta}}$ in (12), we need to know the estimate of the eigenfunction $\overline{\psi}_k(\mathbf{y}_{i,j})$ for all the original training data $\mathbf{y}_{i,j}$. Computationally, however, we can only construct the discrete estimate of the eigenfunction $\overline{\psi}_k(\overline{\mathbf{y}}_b)$ at the reduced box-averaged data points $\overline{\mathbf{y}}_b$. This suggests that we need to extend the eigenfunctions $\overline{\psi}_k(\overline{\mathbf{y}}_b)$ onto all the original training data $\{\mathbf{y}_{i,j}\}_{j=1,\dots,M}^{i=1,\dots,N}$.

For the convenience of discussion, the training data that are used to construct the eigenfunctions are denoted by $\{\mathbf{y}_r^{\text{old}}\}_{r=1,\dots,R}$, and all the data that are not part of $\{\mathbf{y}_r^{\text{old}}\}_{r=1,\dots,R}$ are denoted by $\mathbf{y}^{\text{new}}$.

To extend the eigenfunctions $\overline{\psi}_k\left(\mathbf{y}_r^{\text{old}}\right)$ onto the data point $\mathbf{y}^{\text{new}} \notin \left\{\mathbf{y}_r^{\text{old}}\right\}_{r=1,\ldots,R}$, one approach would be to use the Nyström extension [25] that is based on the basic theory of RKHS [26]. Let $\mathcal{H}_{\bar{q}}\left(\mathcal{N}\right)$ be the RKWHS with a symmetric positive kernel $\widehat{\mathcal{T}} : \mathcal{N} \times \mathcal{N} \to \mathbb{R}$ defined as,

$$\widehat{\mathcal{T}}(\mathbf{y}, \mathbf{y}') = \sum_{k=1}^{\infty} \lambda_k \overline{\psi}_k(\mathbf{y})\overline{\psi}_k(\mathbf{y}'),$$

where λ_k is the corresponding eigenvalue of $\mathcal{L}$ associated with eigenfunction $\overline{\psi}_k$. Then, for any function $f \in \mathcal{H}_{\bar{q}}\left(\mathcal{N}\right)$, the Moore–Aronszajn theorem states that one can evaluate f at $\mathbf{a} \in \mathcal{N}$ with the following inner product, $f(\mathbf{a}) = \left\langle f, \widehat{\mathcal{T}}\left(\mathbf{a}, \cdot\right) \right\rangle_{\mathcal{H}_{\bar{q}}}$. In our application, this amounts to evaluating,

$$\overline{\psi}_k\left(\mathbf{y}^{\text{new}}\right) = \frac{1}{R} \sum_{r=1}^{R} \mathcal{T}\left(\mathbf{y}^{\text{new}}, \mathbf{y}_r^{\text{old}}\right) \overline{\psi}_k\left(\mathbf{y}_r^{\text{old}}\right), \tag{49}$$

where the non-symmetric kernel function $\mathcal{T} : \mathcal{N} \times \mathcal{N} \to \mathbb{R}$ (constructed by the diffusion maps algorithm) is related to the symmetric kernel $\widehat{\mathcal{T}}$ by,

$$\mathcal{T}\left(\mathbf{y}_i, \mathbf{y}_j\right) = \bar{q}^{-1/2}\left(\mathbf{y}_i\right) \widehat{\mathcal{T}}\left(\mathbf{y}_i, \mathbf{y}_j\right) \bar{q}^{1/2}\left(\mathbf{y}_j\right)$$

with $\bar{q}\left(\mathbf{y}_i\right)$ being the sampling density of $\left\{\mathbf{y}_r^{\text{old}}\right\}_{r=1,\ldots,R}$ at $\mathbf{y}_i$. See the detailed evaluation of the kernels $\widehat{\mathcal{T}}$ and $\mathcal{T}$ for the Nyström extension in [27]. After obtaining the estimate of the eigenfunction $\overline{\psi}_k\left(\mathbf{y}^{\text{new}}\right)$ using the Nyström extension, we can train the matrix $\mathbf{C}_{\mathbf{Y}\boldsymbol{\Theta}}$ in (12) for large MN and then obtain the representation of the conditional density on arbitrary new observation $\mathbf{y}_t$, $\widehat{p}\left(\mathbf{y}_t|\boldsymbol{\theta}\right)$.

To summarize this section, we have constructed two different sets of basis functions for $\mathbf{y}$, the analytic basis functions of $L^2\left(\mathcal{N}, q\right)$ such as the Hermite and cosine basis functions, which assume that the manifold is $\mathbb{R}^n$ or hyperrectangle, respectively, and the data-driven basis functions of $L^2\left(\mathcal{N}, \bar{q}\right)$, with $\mathcal{N}$ being the data manifold and $\bar{q}$ being the sampling density that are computed using the VBDM algorithm.

5. Parameter Estimation Using the Metropolis Scheme

First, we briefly review the Metropolis scheme for estimating the posterior density $p(\boldsymbol{\theta}|\mathbf{y}^\dagger)$ given new observations $\mathbf{y}^\dagger = \{\mathbf{y}_1^\dagger, \ldots, \mathbf{y}_T^\dagger\}$ for a specific parameter $\boldsymbol{\theta}^\dagger$. The key idea of the Metropolis scheme is to construct a Markov chain such that it converges to samples of conditional density $p(\boldsymbol{\theta}|\mathbf{y}^\dagger)$ as the target density. In our application, the parameter estimation procedures can be outlined as follows:

(2-A) Suppose we have $\boldsymbol{\theta}_0 \sim p(\boldsymbol{\theta}_0|\mathbf{y}^\dagger) > 0$, then for $i \geq 1$, we can sample $\boldsymbol{\theta}^* \sim \kappa\left(\boldsymbol{\theta}_{i-1}, \boldsymbol{\theta}^*\right)$. Here, κ is the proposal kernel density. For example, use the random walk Metropolis algorithm to generate proposals, $\kappa\left(\boldsymbol{\theta}_{i-1}, \boldsymbol{\theta}^*\right) = \mathcal{N}\left(\boldsymbol{\theta}_{i-1}, \mathbf{C}\right)$, where $\mathbf{C}$, the proposal covariance, and is a tunable nuisance parameter.

(2-B) Accept the proposal, $\boldsymbol{\theta}_i = \boldsymbol{\theta}^*$ with probability $\min\left(\frac{p(\boldsymbol{\theta}^*|\mathbf{y}^\dagger)}{p(\boldsymbol{\theta}_{i-1}|\mathbf{y}^\dagger)}, 1\right)$, otherwise set $\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i-1}$. Repeat Procedures (2-A) and (2-B) above. Notice that the posterior $p(\boldsymbol{\theta}|\mathbf{y}^\dagger)$ can be determined from the prior $p_0(\boldsymbol{\theta})$ and the likelihood $p(\mathbf{y}^\dagger|\boldsymbol{\theta})$ based on Bayes' theorem (3). The likelihood function $p(\mathbf{y}^\dagger|\boldsymbol{\theta})$ is defined as a product of conditional densities of new observations $\mathbf{y}^\dagger = \{\mathbf{y}_1^\dagger, \ldots, \mathbf{y}_T^\dagger\}$ in (44) (Procedure (1-D)). The conditional densities of new observations $\mathbf{y}^\dagger$ given $\boldsymbol{\theta}$ are obtained from the training Procedure (1-C).

(2-C) Generate a sufficiently long chain and use the chain's statistic as an estimator of the true parameter $\boldsymbol{\theta}^\dagger$. Take multiple runs of the chain started at different initial $\boldsymbol{\theta}_0$, and examine whether all these runs converge to the same distribution. The convergence of all the examples below has been validated using 10 randomly-chosen different initial conditions.

In the remainder of this section, we present numerical results of the Metropolis scheme using the proposed data-driven likelihood function on various instructive examples, where the likelihood function is either explicitly known, or can be approximated as in (6), or is intractable. In an example where the explicit likelihood is known, our goal is to show that the approach numerically converges to the true posterior estimate. In the second example, where the dimension of the data manifold is strictly less than the ambient dimension, we will show that the RKHS framework with the knowledge of the intrinsic geometry is superior. When the intrinsic geometrical information is unknown, the proposed data-driven likelihood function is competitive. In the third example with a low-dimensional dynamic and observation model of the form (5), we compare the proposed approach with standard methods, including the direct MCMC and nonintrusive spectral projection (both use the likelihood function of the form (6)). In our last example, we consider an observation model where the likelihood function is intractable and the cost of evaluating the observation model in (4) is numerically expensive.

5.1. Example I: Two-Dimensional Ornstein–Uhlenbeck Process

Consider an Ornstein–Uhlenbeck (OU) process as follows:

$$dX = -\frac{1}{2}X dt + \Sigma^{1/2} dW_t, \tag{50}$$

where $X \equiv (X_1, X_2)^\top$ denotes the state variable, $W_t = (W_1, W_2)^\top$ denotes two-dimensional Wiener processes, and $\Sigma \in \mathbb{R}^{2\times2}$ is a diagonal matrix with main diagonal components $\sigma_{X_1}^2$ and $\sigma_{X_s}^2$ to be estimated. In the stationary process, the solution of Equation (50) $X = (X_1, X_2)^\top$ admits a Gaussian distribution $X \sim \mathcal{N}(0, \Sigma)$,

$$p(X|\Sigma) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left[-\frac{1}{2}X^\top \Sigma^{-1} X\right]. \tag{51}$$

Our goal here is to estimate the posterior density and the posterior mean of the parameters $(\sigma_{X_1}^2, \sigma_{X_2}^2)$, given a finite number, T, of observations, $X^\dagger \equiv (X^{1\dagger}, \ldots, X^{T\dagger})$, for hidden true parameters $((\sigma_{X_1}^2)^\dagger, (\sigma_{X_2}^2)^\dagger) = (6.5, 6.3)$, where each $X^{t\dagger}$ is an i.i.d. sample of (51) for $\Sigma = \Sigma^\dagger$. This example is shown here to verify the validity of the framework of our RKWHS representations for parameter estimation application.

One can show that the likelihood function for this problem is the inverse matrix gamma distribution, $\Sigma \sim IMG\left(\frac{T}{2} - \frac{3}{2}, 2, \Psi\right)$, where $\Psi = X^\dagger(X^\dagger)^\top \in \mathbb{R}^{2\times2}$. If a prior is defined to be also the inverse matrix gamma distribution, $\Sigma \sim IMG(\alpha_0, 2, 0)$, for some value of α_0, then the posterior density $p(\Sigma|X^\dagger)$ can be obtained by applying Bayes' theorem,

$$p(\Sigma|X^\dagger) \sim IMG\left(\alpha_0 + \frac{T}{2}, 2, \Psi\right). \tag{52}$$

The posterior mean can thereafter be obtained as,

$$(\Sigma)_{PM} = \begin{pmatrix} \left(\sigma_{X_1}^2\right)_{PM} & 0 \\ 0 & \left(\sigma_{X_2}^2\right)_{PM} \end{pmatrix} = \frac{\Psi}{T + 2\alpha_0 - 3}. \tag{53}$$

To compare with the analytic conditional density $p(X|\Sigma)$ (51), we trained three RKWHS representations of the conditional density function, $\widehat{p}(X|\Sigma)$, by using the same training dataset. For training, we used $M = 64$ well-sampled uniformly-distributed training parameters (shown in Figure 1), $(\sigma_{X_1}^2, \sigma_{X_2}^2)$, where $\sigma_{X_j}^2 \in \{5, 6, \ldots, 12\}$, which are denoted by $\{\Sigma_j\}_{j=1}^M$. For each training parameter Σ_j, we generated $N = 640{,}000$ well-sampled normally distributed observation data of density in (51) with $\Sigma = \Sigma_j$. For Hermite and cosine representations, we used 20 basis functions

for each coordinate, and then, we could construct $K_1 = 400$ basis functions of two-dimensional observation, **X**, by taking the tensor product. For the VBDM representation, we first reduced the data from $MN = 8 \times 640{,}000$ to $B = B^1 \times B^2 = 100 \times 100$ by the box-averaging method (Appendix A). Subsequently, we trained $K_1 = 400$ data-driven basis functions from the B box-averaged data using the VBDM algorithm [20].

Figure 2a displays the analytic conditional density (51), and Figure 2b–d display the pointwise errors of the conditional densities $\widehat{e}(\mathbf{X}|\mathbf{\Sigma}) \equiv p(\mathbf{X}|\mathbf{\Sigma}) - \widehat{p}(\mathbf{X}|\mathbf{\Sigma})$ for the training parameter $(\sigma^2_{X_1}, \sigma^2_{X_2}) = (5, 5)$. It can be seen from Figures 2b–d that all the pointwise errors are small compared to the analytic $p(\mathbf{X}|\mathbf{\Sigma})$ in Figure 2a, so that all representations of conditional densities $\widehat{p}(\mathbf{X}|\mathbf{\Sigma})$ are in excellent agreement with the analytic $p(\mathbf{X}|\mathbf{\Sigma})$ (Figure 2a). This suggests that for the Hermite representation, the upper bound D (33) in Theorem 1 is finite so that the representation is valid in estimating the conditional density, as can be seen from Figure 2b. On the other hand, the upper bounds D (33) for the cosine and the VBDM representations are always finite, as mentioned in Remark 2 and Proposition 2, respectively. We should also point out that for this example, the VBDM representation performed the worst with errors of order 10^{-4} compared to the Hermite and cosine representations whose errors were on the order of 10^{-6}. This larger error in the VBDM representation was because the data-driven basis functions were estimated by discrete eigenvectors $\overline{\boldsymbol{\psi}}_k \in \mathbb{R}^B$, so additional errors [20] were introduced through this discrete approximation (especially on the high modes) on the box-averaged data, $\{\overline{\mathbf{y}}_b\}_{b=1,\dots,B}$, $B = 10{,}000$. On the other hand, for Hermite and cosine representations, their analytic basis functions are known, so that the errors could be approximated by (25) in Theorem 1.
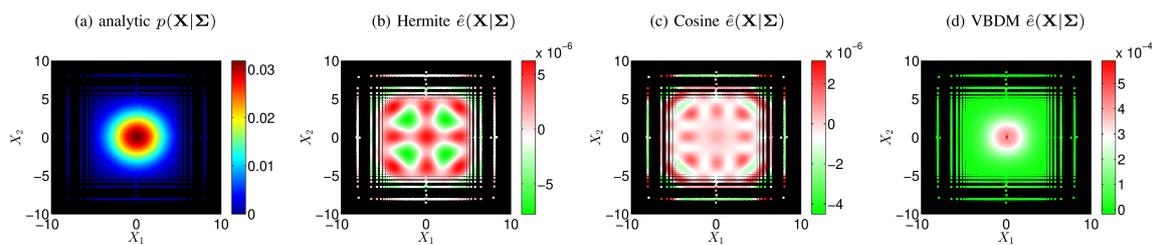

Figure 2. (Color online) (**a**) The analytic conditional density $p(\mathbf{X}|\mathbf{\Sigma})$ (51). For comparison, plotted are the pointwise errors of conditional density functions $\widehat{e}(\mathbf{X}|\mathbf{\Sigma}) \equiv p(\mathbf{X}|\mathbf{\Sigma}) - \widehat{p}(\mathbf{X}|\mathbf{\Sigma})$ for (**b**) Hermite, (**c**) cosine, and (**d**) VBDM representations. The density and all the error functions are plotted on the $B = 10{,}000$ box-averaged data points. The training parameter $\mathbf{\Sigma} \equiv (\sigma^2_{X_1}, \sigma^2_{X_2}) = (5, 5)$.

We now estimate the posterior density (52) and mean (53) by using the MCMC method (Procedures (2-A)–(2-C)). We generated $T = 400$ well-sampled normally-distributed data as the observations from the true values of variance $\mathbf{\Sigma}^\dagger = ((\sigma^2_{X_1})^\dagger, (\sigma^2_{X_2})^\dagger) = (6.5, 6.3)$. From the analytical Formula (53), we obtained the posterior mean as $(\sigma^2_{X_1}, \sigma^2_{X_2})_{PM} = (6.03, 5.84)$. Here, the posterior mean deviated greatly from the true value since we only used $T = 400$ normally-distributed observation data as new observations. If using much more new observation data, the analytical posterior mean (53) will get closer to the true value, $((\sigma^2_{X_1})^\dagger, (\sigma^2_{X_2})^\dagger)$. In our simulation, we set the parameter in the prior, $\alpha_0 = 1$, and the proposal covariance, $\mathbf{C} = 0.01\mathbf{I}$. For each chain, the initial condition $(\sigma^2_{X_1,0}, \sigma^2_{X_2,0})$ was drawn randomly from $U[5, 12]^2$, and 800,000 iterations are generated for the chain.

Figure 3b,c,d display the densities of the chain by using Hermite, cosine, and VBDM representation, respectively. The densities are plotted using the kernel density estimate on the chain ignoring the first 10,000 iterations. For comparison, Figure 3a displays the analytic posterior density (52). It can be seen from Figure 3 that the posterior densities by the three representations were in excellent agreement with each other and with the analytic posterior density (52). Figure 3 also shows the comparison between the posterior mean (53) and the MCMC mean estimates. From our numerical results, MCMC mean estimates by all representations and the analytic posterior mean (53) were identical within numerical

accuracy. Therefore, for this 2D OU-process example, all representations were valid in estimating the posterior density and posterior mean of parameter $\mathbf{\Sigma}$.
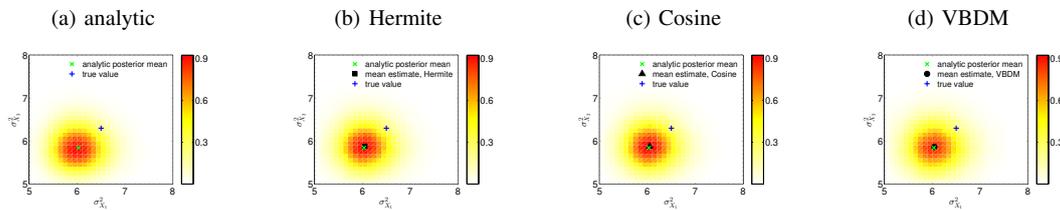

Figure 3. (Color online) Comparison of the posterior density functions $p(\mathbf{\Sigma}|\mathbf{X}^\dagger)$. (a) Analytical posterior density $p(\mathbf{\Sigma}|\mathbf{X}^\dagger)$ (52). (b) Hermite representation. (c) Cosine representation. (d) VBDM representation. The true value $\mathbf{\Sigma}^\dagger \equiv (\sigma^2_{X_1}, \sigma^2_{X_2}) = (6.5, 6.3)$ (blue plus). The analytic posterior mean is $(\sigma^2_{X_1}, \sigma^2_{X_2})_{PM} = (6.03, 5.84)$ (green cross). The MCMC mean estimate using Hermite representation is $(\sigma^2_{X_1}, \sigma^2_{X_2}) = (6.05, 5.87)$ (black square). The MCMC mean estimate using Cosine representation is $(\sigma^2_{X_1}, \sigma^2_{X_2}) = (6.05, 5.87)$ (black triangle). The MCMC mean estimate using VBDM representation is $(\sigma^2_{X_1}, \sigma^2_{X_2}) = (6.04, 5.86)$ (black circle).

Next, we will investigate a system for which the intrinsic dimension d of the data manifold where the observations lie is smaller than the dimension of ambient space n.

5.2. Example II: Three-Dimensional System of SDE's on a Torus

Consider a system of SDE's on a torus defined in the intrinsic coordinates $(\theta, \phi) \in [0, 2\pi)^2$:

$$d \begin{pmatrix} \theta \\ \phi \end{pmatrix} = a\,(\theta, \phi)\,dt + b\,(\theta, \phi) \begin{pmatrix} dW_1 \\ dW_2 \end{pmatrix}, \tag{54}$$

where W_1 and W_2 are two independent Wiener processes, and the drift and diffusion coefficients are:

$$a\,(\theta, \phi) = \begin{pmatrix} \frac{1}{2} + \frac{1}{8} \cos(\theta) \cos(2\phi) + \frac{1}{2} \cos(\theta + \pi/2) \\ 10 + \frac{1}{2} \cos(\theta + \phi/2) + \cos(\theta + \pi/2) \end{pmatrix},$$

$$b\,(\theta, \phi) = \begin{pmatrix} D + D \sin(\theta) & \frac{1}{4} \cos(\theta + \phi) \\ \frac{1}{4} \cos(\theta + \phi) & \frac{1}{40} + \frac{1}{40} \sin(\phi) \cos(\theta) \end{pmatrix}.$$

The initial condition is $(\theta, \phi) = (\pi, \pi)$. Here, D is a parameter to be estimated. This example exhibits non-gradient drift, anisotropic diffusion, and multiple time scales. Both the observations and the training dataset were generated by numerically solving the SDE on appropriate parameters D in (54) with a time step $\Delta t = 0.1$ and then mapping this data into the ambient space, $\mathbb{R}^3$, via the standard embedding of the torus given by:

$$\mathbf{x} \equiv (x, y, z) = ((2 + \sin(\theta)) \cos(\phi), (2 + \sin(\theta)) \sin(\phi), \cos(\theta)). \tag{55}$$

Here, $\mathbf{x} \equiv (x, y, z)$ are observations. This system on a torus satisfies $d < n$, where $d = 2$ is the intrinsic dimension of $\mathbf{x}$ and $n = 3$ is the dimension of ambient space $\mathbb{R}^n$. Our goal is to estimate the posterior density and the posterior mean of parameter D given discrete-time observations of $\mathbf{x}^\dagger$, which are the solutions of (54) for a specific parameter $D^\dagger$.

For training, we used $M = 8$ well-sampled uniformly-distributed training parameters, $\{D_j = j/4\}_{j=1}^8$. For each training parameter D_j, we generated $N = 54{,}000$ observations of $\mathbf{x}$ by solving the SDE's in (54) for parameter D_j. For Hermite and cosine representation, we constructed 10 basis functions for each x, y, z coordinate in Euclidean space. After taking tensor product of these basis functions, we could obtain $K_1 = 1000$ basis functions on the ambient space $\mathbb{R}^3$. For VBDM representation, we first computed $B = B_1 \times B_2 \times B_3 = 30^3$ box-averaged data points by the data

reduction method in Appendix A. However, we found that some of the B box-averaged data points were far away from the torus. After ignoring these points, we eventually chose $\widetilde{B} = 26{,}020$ of the box-averaged data points that were close enough to the torus for training. Then, we trained $K_1 = 1000$ data-driven basis functions on $\mathcal{N}$ from these 26,020 box-averaged data points using the VBDM algorithm.

Unlike the previous example, the derivation of the analytical expression for the likelihood function $p(\mathbf{x}|D_j)$ was not trivial. This difficulty is due to the fact that the diffusion coefficient, $b(\theta, \phi)$, is state dependent. While direct MCMC with an approximate likelihood function constructed using the Bayesian imputation [5] can be done in principle, we neglected this numerical computation since the cost in generating the path $\{\mathbf{x_i}\}$ for $i = 1, \ldots, T$ on each sampling step was too costly in our setup below (where $T = 10{,}000$, and we would generate a chain of length 400,000 samples). For diagnostic comparisons, we constructed another representation $\widehat{p}(\mathbf{x}|D_j)$, named the intrinsic Fourier representation, which can be regarded as an accurate approximation of $p(\mathbf{x}|D_j)$, as it used the basis functions defined on the intrinsic coordinates (θ, ϕ) instead of $\mathbf{x} \in \mathbb{R}^3$. See Appendix B for the construction and the convergence of the intrinsic Fourier representation in detail. We should point out that this intrinsic representation is not available in general since one may not know the embedding of the data manifold.

Figure 4 displays the comparison of the density estimates. It can be observed from Figure 4 that the VBDM representation was in good agreement with the intrinsic Fourier representation, whereas Hermite and cosine representations of $\widehat{p}(\mathbf{x}|D_j)$ deviated significantly from the intrinsic Fourier representation. The reason in short was that if the density $p(\theta, \phi|D)$ in (θ, ϕ) coordinate were in $\mathcal{H}([0, 2\pi)^2) \subset L^2([0, 2\pi)^2)$, then the corresponding VBDM representation with respect to $dV(\mathbf{x})$ would be in $\mathcal{H}_{\bar{q}^{-1}}(\mathcal{N})$. However, the representation (Hermite and cosine) with respect to $d\mathbf{x}, \mathbf{x} \in \mathbb{R}^3$ is not in $\mathcal{H}_{\bar{q}^{-1}}(\mathbb{R}^3)$. A more detailed explanation of this assertion is presented in Appendix B.

We now compare the MCMC estimates with the true value, $D^\dagger = 0.9$, from $T = 10{,}000$ observations. For this simulation, we set the prior to be uniformly distributed and empirically chose $C = 0.01$ for the proposal. Figure 5 displays the posterior densities of the chains for all representations (each plot of the density estimate was constructed using KDE on a chain of length 400,000). Displayed also is the comparison between the true value $D^\dagger$ and the MCMC mean estimates by all representations. Here, the mean estimate by the intrinsic Fourier representation nearly overlaps with the true value $D^\dagger = 0.9$, as shown in Figure 5. The mean estimate by the VBDM representation is closer to the true value $D^\dagger$ compared to the estimates by Hermite and cosine representations. Moreover, it can be seen from Figure 5 that the posterior by the VBDM representation is close to the posterior by intrinsic Fourier representation, whereas the posterior densities by Hermite and cosine representation are not. We should point out that this result is encouraging considering that the training parameter domain is rather wide, $D_j \in [1/4, 2]$. This result suggests that when the intrinsic dimension is less than the ambient space dimension, $d < n$, the VBDM representation (which does not require the knowledge of the embedding function in (55)) with data-driven basis functions in $L^2(\mathcal{N}, \bar{q})$ is superior compared to the representations with analytic basis functions defined on the ambient coordinates $\mathbb{R}^3$.
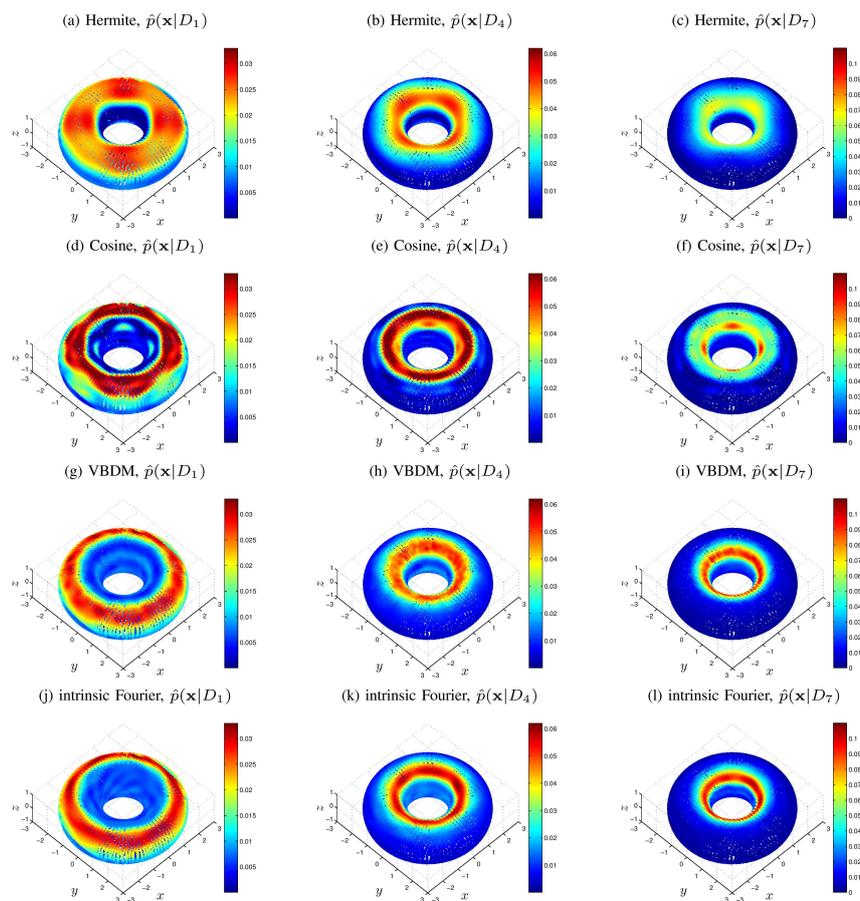
Figure 4. (Color online) Comparison of the conditional densities $\hat{p}(\mathbf{x}|D_j)$ estimated by using Hermite representation (first row), cosine representation (second row), VBDM representation (third row), and intrinsic Fourier representation (fourth row). The left (a,d,g,j), middle (b,e,h,k), and right (c,f,i,l) columns correspond to the densities on the training parameters $D_1 = 0.25$, $D_4 = 1.00$, and $D_7 = 1.75$, respectively. $K_1 = 1000$ basis functions are used for all representations. For fair visual comparison, all conditional densities are plotted on the same box-averaged data points and normalized to satisfy $\frac{1}{\tilde{B}}\sum_{b=1}^{\tilde{B}}\hat{p}(\mathbf{x}_b|D_j)/\bar{q}(\mathbf{x}_b) = 1$ with $\bar{q}$ being the estimated sampling density of the box-averaged data $\{\mathbf{x}_b\}_{b=1}^{\tilde{B}}$.
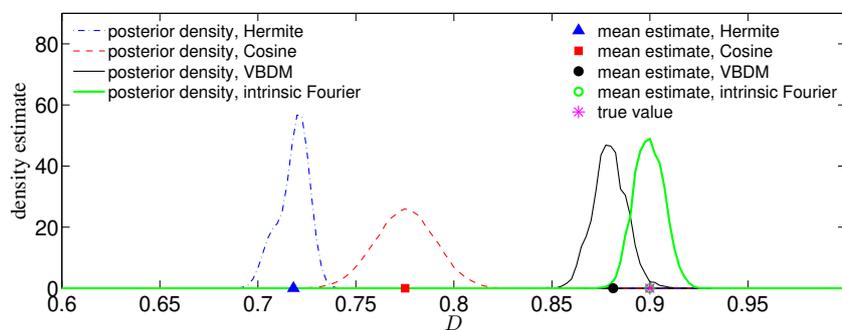
Figure 5. (Color online) Comparison of the posterior density functions by all representations. Plotted also are mean estimates by Hermite representation $\hat{D} = 0.78$ (blue triangle), cosine representation $\hat{D} = 0.79$ (red square), VBDM representation $\hat{D} = 0.88$ (black circle), the intrinsic Fourier representation $\hat{D} = 0.90$ (green circle), and the true parameter value $D^{\dagger} = 0.9$ (magenta asterisk).

5.3. Example III: Five-Dimensional Lorenz-96 Model

Consider the Lorenz-96 model [28]:

$$\frac{dx_j}{dt} = x_{j-1}\left(x_{j+1} - x_{j-2}\right) - x_j + F, \quad j = 1, \ldots, J, \tag{56}$$

with periodic boundary, $x_{j+J} = x_j$. For the example in this section, we set $J = 5$. The initial condition was $x_j(0) = \sin(2\pi j/5)$. Our goal here was to estimate the posterior density and posterior mean of the hidden parameter F given a time series of noisy observations $\mathbf{y}^\dagger = \left(y_1^\dagger, y_2^\dagger, y_3^\dagger, y_4^\dagger, y_5^\dagger\right)$, where:

$$y_j^\dagger(t_m) = x_j^\dagger(t_m) + \epsilon_{m,j}, \quad \epsilon_{m,j} \sim \mathcal{N}\left(0, \sigma^2\right), \quad m = 1, \ldots, T,$$

with noise variance $\sigma^2 = 0.01$. Here, $x_j^\dagger(t_m)$ denotes the approximate solution (with the Runge–Kutta method) with a specific parameter value $F^\dagger$ at discrete times $t_m = ms\triangle t$, where $\triangle t = 0.05$ is the integration time step and s is the observation interval. Since the embedding function of the observation data is unknown, we do not have a parametric analog to the intrinsic Fourier representation as in the previous example.

In this low-dimensional setting, we can compare the proposed method with basic techniques, including the direct MCMC and the Non-Intrusive Spectral Projection (NISP) method [13]. By direct MCMC, we refer to employing the random walk Metropolis scheme directly on the following likelihood function,

$$p\left(\mathbf{y}^\dagger | F\right) \propto \exp\left(-\frac{\sum_{m=1}^{T}\sum_{j=1}^{5}\left(y_j^\dagger(t_m) - x_j(t_m; F)\right)^2}{2\sigma^2}\right), \tag{57}$$

where σ^2 is the noise variance and $x_j(t_m; F)$ is the solution of the initial value problem in Equation (56) with the parameter F at time t_m. Note that evaluating $x_j(t_m; F)$ is time consuming if the model time $Ts\Delta t$ is long or the MCMC chain has many iterations. In our implementation, we generated the chain for 4000 iterations. This amounts to 4000 sequential evaluations of the likelihood function in (57), where each evaluation requires integrating the model in (56) with the proposal parameter value F^* until model unit time $Ts\Delta t$. We used a uniform prior distribution and $C = 0.1$ for the proposal.

For the NISP method [13], we used the same Gaussian likelihood function (57) with approximated x_j. In particular, we approximated the solutions x_j with $\tilde{x}_j(t, F)$ for $j = 1, \ldots, 5$ in the form of:

$$\tilde{x}_j(t, F) = \sum_{k=1}^{K} \hat{x}_{j,k}(t)\, \varphi_k(F), \tag{58}$$

where $\varphi_k(F)$ are chosen to be the orthonormal cosine basis functions, $\hat{x}_{j,k}(t)$ are the expansion coefficients, and K is the number of basis functions. Subsequently, we prescribe a fixed set of nodes $\left\{F_j = 7.55 + 0.1j\right\}_{j=1}^{8}$ to be used for training $\hat{x}_{j,k}(t)$. Practically, this training procedure only requires eight model evaluations that can be done in parallel, where each evaluation involves integrating the model with the specified F_j until model unit time $Ts\Delta t$. The number of basis functions is $K = 8$. After specifying the coefficients $\hat{x}_{j,k}(t)$ such that $\tilde{x}_j(t, F) = x_j(t; F)$, we obtain the approximation of the solutions $\tilde{x}_j(t, F)$ for all parameters F. Using these approximated $\tilde{x}_j(t, F)$, in place of $x_j(t_m, F)$ in (57), we can generate the Markov chain using the Metropolis scheme. Again, we used a uniform prior distribution and $C = 0.1$ for the proposal. In our MCMC implementation, we generated the chain for 40,000 iterations; this involved only evaluating (58) instead of integrating the true dynamical model in (56) on the proposal parameter value F^*.

For RKWHS representations, we also used $M = 8$ uniformly-distributed training parameters, $\left\{F_j = 7.55 + 0.1j\right\}_{j=1}^{8}$. As in the NISP, this training procedure required only eight model integrations

with parameter value F_j until the model unit time $Ts\Delta t$, resulting in a total of $MN = 8Ts$ training data. In this example, we did not reduce the data using the box-averaging method in Appendix A. In fact, for some cases, such as $s = 1$ and $T = 50$, the total of training data were only $MN = 400$, which was too few for estimation of the eigenfunctions. Of course, one can consider more training parameters to increase this training dataset, but for a fair comparison with NISP, we chose to just add 10 i.i.d. Gaussian noises to each dataset, resulting in a total of $MN = 4000$ for training dataset. This configuration (with a small dataset) is a tough setting for the VBDM since the nonparametric method is advantageous in the limit of a large dataset. When $8Ts$ is sufficiently large, we do not need to increase the dataset by adding multiple i.i.d Gaussian noises.

For Hermite and Cosine representation, we constructed five Hermite and cosine basis functions for each coordinate, which yielding a total of $K_1 = 5^5 = 3125$ basis functions in $\mathbb{R}^5$. For the VBDM representation, we directly applied the VBDM algorithm to train $K_1 = 3125$ data-driven basis functions on manifold $\mathcal{N}$ from the $MN = 4000$ training dataset. From the VBDM algorithm, the estimated intrinsic dimension was $d \approx 2$, which was smaller than the dimension of the ambient space $n = 5$. Then, we applied a uniform prior distribution and $C = 0.01$ for the proposal. As in NISP, we generated the chain for 40,000 iterations, which amounted to evaluating (44) instead of integrating the true dynamical model in (56) on each iteration.

We now compare the posterior densities and mean estimates for the case of $s = 1$ and $T = 50$ noisy observations $\mathbf{y}^\dagger (t_m)$ corresponding to the true parameter value $F^\dagger = 8$. Figure 6 displays the posterior densities of the chains and mean estimates for the direct MCMC method, NISP method, and all representations. It can be seen from Figure 6 that the mean estimate by VBDM representation was in good agreement with the true value $F^\dagger$. In contrast, the mean estimates by Hermite and cosine representations deviated substantially from the true value. Based on this numerical result, where the estimated intrinsic dimension $d \approx 2$ of the observations was lower than the ambient space dimension $n = 5$, the data-driven VBDM representation was superior compared to the Hermite and cosine representations. It can be further observed that direct MCMC, NISP, and VBDM representation can provide good mean estimates to the true value. However, notice that we only ran the model $M = 8$ times for the NISP method and VBDM representation, whereas we ran the model 4000 times for the direct MCMC method.
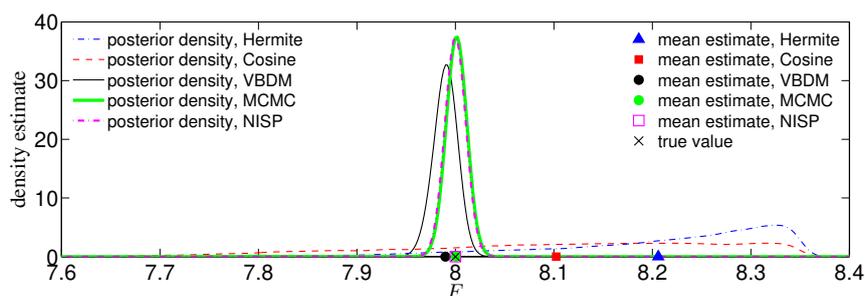

Figure 6. (Color online) Comparison of the posterior density functions among the direct MCMC method, NISP method, and all RKWHS representations. Plotted also are the true parameter value $F^\dagger = 8$ (black cross), mean the estimate by the direct MCMC method $\hat{F} = 8.00$ (green circle), the mean estimate by the NISP method $\hat{F} = 8.00$ (magenta square), and the mean estimates by Hermite representation $\hat{F} = 8.21$ (blue triangle), cosine representation $\hat{F} = 8.10$ (red square), and VBDM representation $\hat{F} = 7.99$ (black circle). The noisy observations are $y_{j^\dagger}(t_m)$ for $s = 1$, $T = 50$.

In real applications where the observations are not simulated by the model, we expect the observation configuration to be pre-determined. Therefore it is important to have an algorithm that is robust under various observation configurations. In our next numerical experiment, we checked such robustness by comparing the direct MCMC method, NISP method, and VBDM representation for different cases of s and T (Figure 7a). It can be observed from Figure 7a that both the direct MCMC

method and VBDM representation can provide reasonably accurate mean estimates for all cases of s and T. However, again notice that we need to run the model much more times for the direct MCMC method than for VBDM representation. It can be further observed that the NISP method can only provide a good mean estimate for observation time up to $Ts\Delta t = 200\Delta t$ when eight uniform nodes $\{F_j = 7.55 + 0.1j\}_{j=1}^{8}$ are used. The reason was that the approximated solution by NISP method was only accurate for observation time up to $200\Delta t$ (see the green and red curves in Figure 7b). This result suggests that our surrogate modeling approach using the VBDM representation can provide accurate and robust mean estimates under various observation configurations.

(a)

(b)
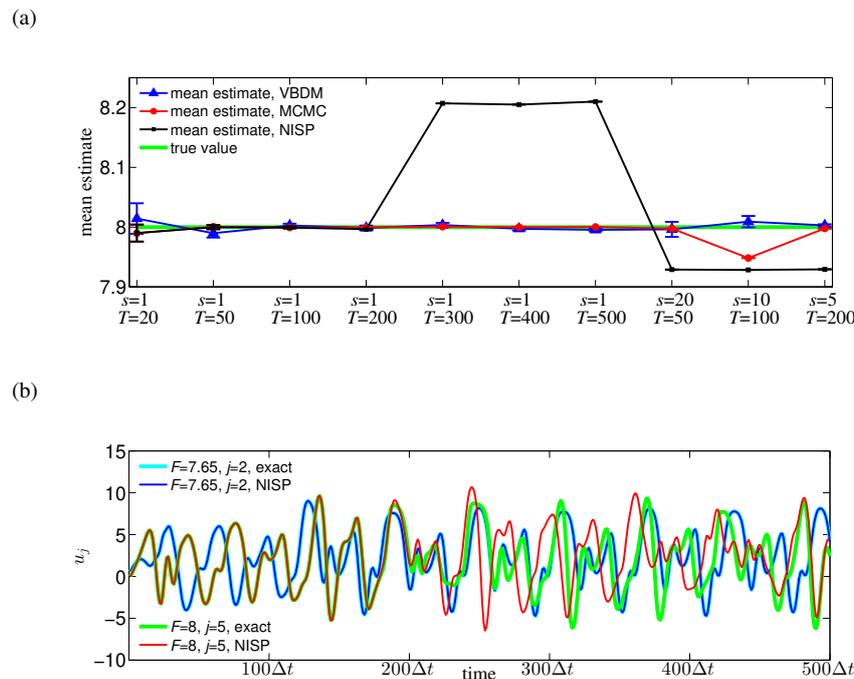
Figure 7. (Color online) (**a**) Comparison of the mean estimates among the direct MCMC method, NISP method, and VBDM representation for different cases of s and T. Plotted also is the true parameter value $F^\dagger = 8$ (green curve). (**b**) Comparison of the exact solution by numerical integration and the approximated solution by the NISP method at the training parameter $F = 7.65$ and at the parameter value $F = 8$, which is not in the training parameter.

5.4. Example IV: The 40-Dimensional Lorenz-96 Model

In this section, we consider estimating the parameter F in the Lorenz-96 model in (56), but of a $J = 40$ dimensional system. We now consider observing the autocorrelation function of several energetic Fourier modes of the system phase-space variables. In particular, let $\{\hat{x}_k(t_m; F)\}_{k=-J/2+1,\ldots,J/2}$ be the kth discrete Fourier mode of $\{x_j(t_m; F)\}_{k=1\ldots,J}$, where $t_m = m\Delta t$ with $\Delta t = 0.05$. Let the observation function be defined as in (4) with four-dimensional $\{\mathbf{y}_m(F)\}_{m=0,\ldots,T}$, whose components are the autocorrelation function of Fourier mode k_j,

$$y_{m,j}(F) = \mathbb{E}[\hat{x}_{k_j}(t_m; F)\hat{x}_{k_j}(t_0; F)], \quad m = 0,\ldots,T, \quad j = 1,\ldots,4,$$

of the energetic Fourier modes, $k_j \in \{7, 8, 9, 14\}$. See [29] for the detailed discussion of the statistical equilibrium behavior of this model for various values of F. Such observations arise naturally since some of the model parameters can be identified from non-equilibrium statistical information via the

linear response statistics [30,31]. In our numerics, we will approximate the correlation function by averaging over a long trajectory,

$$\mathbb{E}[\hat{x}_{k_j}(t_m; F)\hat{x}_{k_j}(t_0; F)] \approx \frac{1}{L} \sum_{\ell=1}^{L} \hat{x}_{k_j}(t_{m+\ell}; F)\hat{x}_{k_j}(t_\ell; F), \qquad (59)$$

with $L = 10^6$. Here, each of these Fourier modes is assumed to have zero empirical mean. We will consider observing the autocorrelation function up to time $T = 50$ (corresponding to 2.5 unit time).

　　With this setup, the corresponding likelihood function for $p(\mathbf{y}_m | F)$ is not easily approximated (since it is not in the form of (6)), and it is computationally demanding to generate $y_{m,j}(F)$ since each evaluation requires integration of the 40-dimensional Lorenz-96 model up to time index $L = 10^6$. This expensive computational cost makes either the direct MCMC or approximate Bayesian computation infeasible. We should also point out the fact that a long trajectory is needed in the evaluation of (59), making this problem intractable with NISP even if a parametric likelihood function becomes available. This issue is because the approximated trajectory by polynomial chaos expansion in NISP is only accurate for short times, as shown in the previous example. We will consider constructing the likelihood function from a wide range of training parameter values, $F_i = 6 + 0.1(i - 1), i = 1, \dots, M = 31$. This parameter domain is rather wide and includes the weakly chaotic regime ($F = 6$) and strongly chaotic regime ($F = 8$). See [32] for a complete list of chaotic measures in these regimes including the largest Lyapunov exponent and the Kolmogorov–Sinai entropy.

　　In this setup, we had a total of $MN = M(T + 1) = 31 \times 51 = 1581$ of $\mathbf{y}_m(F_i) \in \mathbb{R}^4$ for training. We will consider an RKHS representation with $K_1 = 500$ basis functions. We will demonstrate the performance on 30 sets of observations $\mathbf{y}_m(F_s^\dagger)$, where in each case, $F_s^\dagger$ does not belong to the training parameter set, namely $F_s^\dagger = 6.05 + 0.1(s - 1), s = 1, \dots, 30$. In each simulation, the MCMC initial chain will be set to be random, $F \sim \mathcal{U}(6.5, 8.5)$; the prior is uniform; and $C = 0.01$ for the proposal. In Figure 8, we show the mean estimates and an error bar (based on one standard deviation) computed from averaging the MCMC chain of length 40,000 in each case. Notice the robustness of these estimates on a wide range of true parameter values $F^\dagger$ using a likelihood function constructed using a single set of training parameter values on $[6, 9]$.
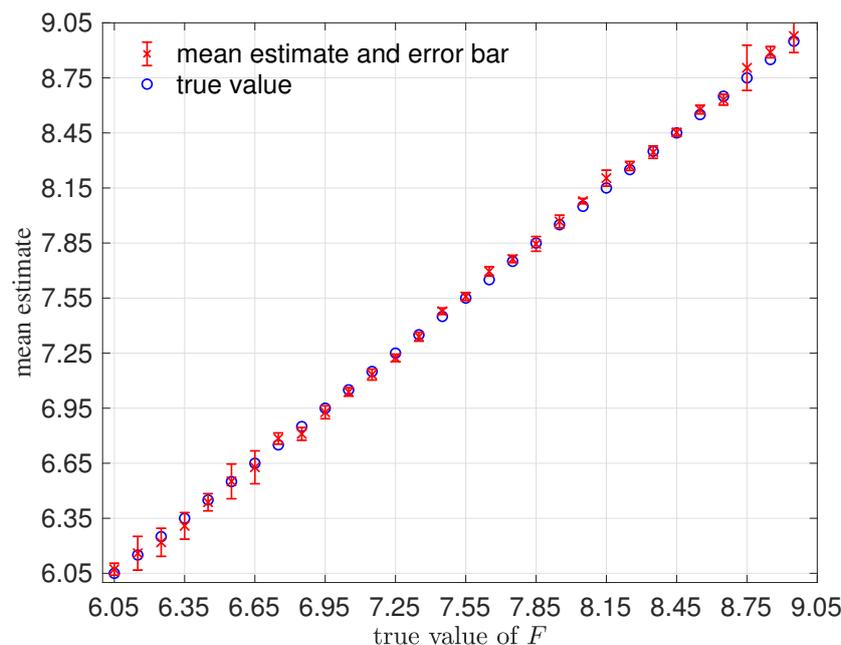

Figure 8. (Color online) Mean error estimates and error bars for various true values of F that are not in the training parameters.

6. Conclusions

We have developed a framework of a parameter estimation approach where MCMC was employed with a nonparametric likelihood function. Our approach approximated the likelihood function using the kernel embedding of conditional distribution formulation based on RKWHS. By analyzing the error estimation in Theorem 1, we have verified the validity of our RKWHS representation of the conditional density as long as $p(\mathbf{y}|\boldsymbol{\theta}_j) \in \mathcal{H}_{q^{-1}}(\mathcal{N})$ induced by the basis in $L^2(\mathcal{N}, q)$ and $\text{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}[\psi_k(\mathbf{Y})]$ is finite. Furthermore, the analysis suggests that if the weight q is chosen to be the sampling density of the data, the $\text{Var}_{\mathbf{Y}|\boldsymbol{\theta}_j}[\psi_k(\mathbf{Y})]$ is always finite. This justifies the use of Variable Bandwidth Diffusion Maps (VBDM) for estimating the data-driven basis functions of the Hilbert space weighted by the sampling density on the data manifold.

We have demonstrated the proposed approach with four numerical examples. In the first example, where the dimension of the data manifold was exactly the dimension of the ambient space, $d = n$, the RKHS representation with VBDM basis yielded a parameter estimate as accurate as using other analytic basis representation. However, in the examples where the dimension of the data manifold was strictly less than the dimension of the ambient space, $d < n$, only VBDM representation could provide more accurate estimation of the true parameter value. We also found that VBDM representation produced mean estimates that were robustly accurate (with accuracies that were comparable to the direct MCMC) on various observation configurations where the NISP was not accurate. This numerical comparison was based on using only eight model evaluations, which can be done in parallel for both VBDM and NISP, whereas the direct MCMC involved 4000 sequential model evaluations. Finally, we demonstrated robust accurate parameter estimation on an example where the analytic likelihood function was intractable and computationally demanding, even if it became available. Most importantly, this result was based on training on a wide parameter domain that included different chaotic dynamical behaviors.

From our numerical experiments, we conclude that the proposed nonparametric representation was advantageous in any of these configurations: (1) when the parametric likelihood function was not known, such as in Example IV; (2) when the observation time stamp was long (such as in Example II or for large sT in Example III and Example IV). Ultimately, the only real advantage of this method (as a surrogate model) was when the direct MCMC or ABC, which require sequential model evaluation, was computationally not feasible.

While the theoretical and numerical results were encouraging as a proof the concept for using the VBDM representation in many other parameter estimation applications, there were still practical limitations that need to be overcome. As in the other surrogate modeling approaches, one needs to have knowledge of the feasible domain for the parameters. Even when the parameter domain is given and wide, it is practically not feasible to generate training dataset by evaluating the model on the specified training grid points on this domain when the dimension of the parameter space is large (e.g., order 10), even if the Smolyak sparse grid is used. One possible way to simultaneously overcome these two issues is to use "crude" methods, such as ensemble Kalman filtering or smoothing, to obtain the training parameters. We refer to such a method as "crude" since the parameter estimation with ensemble Kalman filtering is sensitive to the initial conditions, especially when the persistent model is used as the dynamical model for the parameters [23]. However, with such crude methods, we can at least obtain a set of parameters that reflect the observational data, instead of specifying training parameters uniformly or in a random fashion, which can lead to unphysical training parameters. Another issue that arises in the VBDM representation is the expensive computational cost when the amount of data MN is large. When the dimension of the observations is low (as in the examples in this paper), the data reduction technique described in Appendix A is sufficient. For larger dimensional problems, a more sophisticated data reduction is needed. Alternatively, one can explore representations using other orthonormal data-driven basis, such as the QR factorized basis functions as a less expensive alternative to the eigenbasis [27].

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; nor in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

VBDM   Variable Bandwidth Diffusion Maps
RKHS   Reproducing Kernel Hilbert Space
RKWHS   Reproducing Kernel Weighted Hilbert Space
MCMC   Markov Chain Monte Carlo
ABC   Approximate Bayesian Computation
NISP   Non-Intrusive Spectral Projection

Appendix A. Data Reduction

When MN is very large, the VBDM algorithm becomes numerically expensive since it involves solving an eigenvalue problem of matrix size $MN \times MN$. Notice that the number of training parameters M grows exponentially as a function of the dimension of parameter, m, if well-sampled uniformly distributed training parameters are used. To overcome this large training data problem, we employ an empirical data reduction method to reduce the original MN training data points $\{\mathbf{y}_{i,j}\}_{j=1,\ldots,M}^{i=1,\ldots,N}$ to a small $B (\ll MN)$ number of training data points yet preserving the sampling density $\bar{q}(\mathbf{y})$ in (37). Subsequently, we apply the VBDM algorithm on these reduced training data points. It is worthwhile to mention that this data reduction method is numerically applicable for low-dimensional dataset although in the following we will introduce this reduction method for any n-dimensional dataset.

The basic idea of our method is that we first cluster the training dataset $\{\mathbf{y}_{i,j}\}_{j=1,\ldots,M}^{i=1,\ldots,N}$ into B number of boxes and then take the average of data points in each box as a reduced training data point. First, we cluster the training data $\{\mathbf{y}_{i,j}\}$, based on the ascending order of the 1st coordinate of the training data, $\{y_{i,j}^1\}$, into B^1 number of groups such that each group has the same number $(= MN/B^1)$ of data points. After the first clustering, we obtain B^1 groups with each group denoted by $G_{k_1}^1$ for $k_1 = 1, \ldots, B^1$. Here, the super-index 1 denotes the first clustering and the sub-index k_1 denotes the k_1^{th} group. Second, for each group $G_{k_1}^1$, we cluster the training data $\{\mathbf{y}_{i,j}\}$ inside $G_{k_1}^1$, based on the ascending order of the second coordinate of the training data, $\{y_{i,j}^2\}$, into B^2 number of groups such that each group has the same number $(= MN/B^1B^2)$ of data points. After the second clustering, we obtain totally B^1B^2 groups with each group denoted by $G_{k_1k_2}^2$ for $k_1 = 1, \ldots, B^1$, and $k_2 = 1, \ldots, B^2$. We can operate such clustering n times, where n is the dimension of the observation $\mathbf{y}$ ambient space. After n times clustering, we obtain $B \equiv \prod_{s=1}^n B^s$ groups with each group denoted by $G_{k_1k_2\ldots k_n}^n$ with $k_s = 1, \ldots, B^s$, for all $s = 1, \ldots, n$. Each group is a box [see Figure A1 for example]. After taking the average of the data points in each box $G_{k_1k_2\ldots k_n}^n$, we obtain B number of reduced training data points. In the remainder of this paper, we denote these B number of reduced training data points by $\{\bar{\mathbf{y}}_b\}_{b=1,\ldots,B}$ and refer to them as the box-averaged data points. Intuitively, this algorithm partitions the domain into hyperrectangle such that $P_r(\bar{\mathbf{y}} \in G_{k_1\ldots k_n}^n) \approx 1/B$. Note that the idea of our data reduction method is analogous to that of multivariate k-nearest neighbor density estimates [33,34]. The error estimation can be found in the Refs. [33,34].

To examine whether the distribution of these box-averaged data points is close to the sampling density $\bar{q}(\mathbf{y})$ of the original dataset, we apply our reduction method to several numerical examples in the following. Figure A1a shows the reduction result for few number ($= 64$) of uniform data in $[0,1] \times [0,1]$. Here, $B^1 = 4$ and $B^2 = 4$ so that there are total $B = 16$ boxes and inside each box there are 4 uniform data points (blue circles). It can be seen that the box-averaged data points (red circles) are far away from the well-sampled uniform data points (cyan crosses). However, when uniform data points (blue circles) increase to a large number ($= 6400$), box-average data points (red circles) are very close to well-sampled uniform data points (cyan crosses) as shown in Figure A1b. This suggests that these box-averaged data points nearly admit the uniform distribution when there are a large number of original uniform data points. Figures A1c and d show the comparison of the kernel density estimates applied on the box-averaged data for different B for standard normal distribution and the distribution proportional to $\exp\left[-\left(-X_1^2 + X_1^3 + X_1^4\right)\right]$, respectively. It can be seen that the reduced box-averaged data points nearly preserve the distribution of the original large dataset, $N = 640,000$.
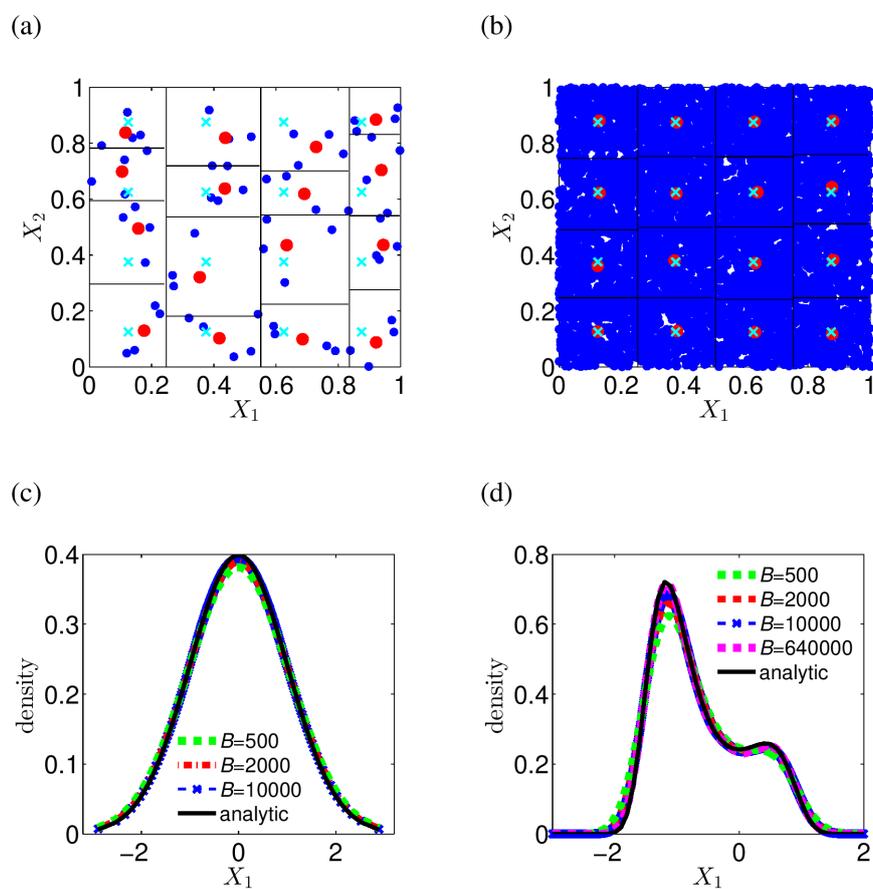

Figure A1. (Color online) Data reduction for (**a**) few number ($= 64$) of uniformly distributed data, and (**b**) many number ($= 6400$) of uniformly distributed data. The 64 blue circles correspond to uniformly distributed data, 16 cyan crosses correspond to well-sampled uniformly distributed data, and 16 red circles correspond to box-averaged data. Boxes are partitioned by horizontal and vertical black lines. The vertical black lines correspond to the first clustering and the horizontal lines correspond to the second clustering. Panels (**c**) and (**d**) display the comparison of kernel density estimates on the box-averaged data for different number B for (c) standard normal distribution, and (d) the distribution proportional to $\exp[-(-X_1^2 + X_1^3 + X_1^4)]$, respectively. For comparison, also plotted is the analytic probability density of the distribution. The total number of the points is $640,000$. It can be seen that the reduced box-averaged data points nearly preserve the distribution of the original dataset.

When *MN* is very large, the VBDM algorithm for the construction of data-driven basis functions [Procedure (*1-B*)] can be outlined as follows. We first use our data reduction method to obtain $B (\ll MN)$ number of box-averaged data $\{\bar{\mathbf{y}}_b\}_{b=1,...,B} \subseteq \mathcal{N} \subseteq \mathbb{R}^n$ with sampling density $\bar{q}(\mathbf{y}) \approx \frac{1}{M}\sum_{j=1}^{M} p(\mathbf{y}|\boldsymbol{\theta}_j)$ in (37). The sampling density $\bar{q}$ is estimated at the box-averaged data $\bar{\mathbf{y}}_b$ using all the box-averaged data points $\{\bar{\mathbf{y}}_b\}_{b=1,...,B}$ by a kernel density estimation method. Implementing the VBDM algorithm, we can obtain orthonormal eigenvectors $\overline{\boldsymbol{\psi}}_k \in \mathbb{R}^B$, which are discrete estimates of the eigenfunctions $\overline{\psi}_k(\mathbf{y}) \in L^2(\mathcal{N}, \bar{q})$. The bth component of the eigenvector $\overline{\boldsymbol{\psi}}_k$ is a discrete estimate of the eigenfunction $\overline{\psi}_k(\bar{\mathbf{y}}_b)$, evaluated at the box-averaged data point $\bar{\mathbf{y}}_b$. Due to the dramatic reduction of the training data, the computation of these eigenvectors $\overline{\boldsymbol{\psi}}_k \in \mathbb{R}^B$ becomes much cheaper than the computation of the eigenvectors $\overline{\boldsymbol{\psi}}_k \in \mathbb{R}^{MN}$ using the original training dataset $\{\mathbf{y}_{i,j}\}_{j=1,...,M}^{i=1,...,N}$. Then we can obtain a discrete representation (10) of the conditional density at the box-averaged data points $\bar{\mathbf{y}}_b$, $\hat{p}(\bar{\mathbf{y}}_b|\boldsymbol{\theta})$.

Appendix B. Additional Results on Example II

In this section, we discuss the intrinsic Fourier representation constructed for numerical comparisons in Example II and provide more detailed discussion on the numerical results.

We first discuss the construction for the intrinsic Fourier representation of the true conditional density, $p(\mathbf{x}|D)$, defined with respect to the volume form inherited by $\mathcal{N}$ from the ambient space $\mathbb{R}^n$ for the system on the torus (54) in Example II. By noticing the embedding (55) of (θ, ϕ) in $\mathbf{x} \equiv (x, y, z)$, we can obtain the following equality,

$$
\begin{aligned}
1 &= \int_{\mathcal{N}} p(\mathbf{x}|D)\, dV(\mathbf{x}) = \int_{[0,2\pi)^2} p(\mathbf{x}(\theta,\phi)|D)\, |\mathbf{x}_\theta \times \mathbf{x}_\phi|\, d\theta d\phi \\
&\equiv \int_{[0,2\pi)^2} p_{\text{IC}}(\theta,\phi|D)\, d\theta d\phi,
\end{aligned} \tag{A1}
$$

where $dV(\mathbf{x}) = |\mathbf{x}_\theta \times \mathbf{x}_\phi|\, d\theta d\phi$ is the volume form, p_{IC} denotes the true conditional density as a function of the intrinsic coordinates, (θ, ϕ). Assuming that $p_{\text{IC}}(\theta,\phi|D) \in \mathcal{H}([0,2\pi)^2) \subset L^2([0,2\pi)^2)$, and the relation in (A1), we can construct the intrinsic Fourier representation as follows,

$$
\hat{p}(\mathbf{x}|D) = \frac{\hat{p}_{\text{IC}}(\theta,\phi|D)}{|\mathbf{x}_\theta \times \mathbf{x}_\phi|}, \tag{A2}
$$

where $\hat{p}_{\text{IC}}(\theta,\phi|D)$ is a RKWHS representation (10) of the conditional density $p_{\text{IC}}(\theta,\phi|D)$ with a set of orthonormal Fourier basis functions $\psi_k(\theta,\phi) \in L^2([0,2\pi)^2)$. Here, $\psi_k(\theta,\phi)$ are formed by the tensor product of two sets of orthonormal Fourier basis functions $\left\{1, \left\{\sqrt{2}\cos(m\theta)\right\}, \left\{\sqrt{2}\sin(m\theta)\right\}\right\}$ and $\left\{1, \left\{\sqrt{2}\cos(m\phi)\right\}, \left\{\sqrt{2}\sin(m\phi)\right\}\right\}$ for $m \in \mathbb{N}^+$. Note that for intrinsic Fourier representation, we need to know the embedding (55) and know the data of (θ, ϕ) in intrinsic coordinates for training, which is available for this example. Nevertheless, for Hermite, Cosine, and VBDM representations, we only need to know the observation data $\mathbf{x}$ for training.

The convergence of $\hat{p}(\mathbf{x}|D)$ to the true density can be explained as follows. For the system (54) in the intrinsic coordinates (θ, ϕ), where $p_{\text{IC}}(\theta,\phi|D) \in \mathcal{H}([0,2\pi)^2)$ for all parameter D, the statistics $\text{Var}_{(\theta,\phi)|D}[\psi_k(\theta,\phi)]$ are bounded for all D and all $k \in \mathbb{N}^+$, by the compactness of $[0,2\pi)^2$ and the uniform boundedness of $\psi_k(\theta,\phi)$ for all k. According to Theorem 1, we can obtain the convergence of the representation $\hat{p}_{\text{IC}}(\theta,\phi|D)$. Then by noticing the smoothness of $|\mathbf{x}_\theta \times \mathbf{x}_\phi|$ on the torus, we can obtain the convergence of the intrinsic Fourier representation $\hat{p}(\mathbf{x}|D)$ in (A2).

Next, we give an intuitive explanation for the reason why in the regime $d < n$, VBDM representation can provide a good approximation whereas Hermite and Cosine representations cannot. Essentially, the VBDM representation uses basis functions of the weighted Hilbert space of functions

defined with respect to a volume form $\widetilde{V}$ that is conformally equivalent to the volume form V that is inherited by the data manifold $\mathcal{N}$ from the ambient space, $\mathbb{R}^n$. That is, the weighted Hilbert space, $L^2(\mathcal{N}, \bar{q}^{-1})$, means,

$$L^2(\mathcal{N}, \bar{q}^{-1}) = \left\{ f : \int_{\mathcal{N}} |f(\mathbf{x})|^2 d\widetilde{V}(\mathbf{x}) < \infty \right\},$$

where $d\widetilde{V}(\mathbf{x}) = \bar{q}(\mathbf{x})^{-1} dV(\mathbf{x})$ denotes the volume form that is conformally changed by the sampling density $\bar{q}$. We should point out that the key point of the diffusion maps algorithm [19] is to introduce an appropriate normalization to avoid biased in the geometry induced by the sampling density $\bar{q}$ when the data are not sampled according to the Riemannian metric inherited by $\mathcal{N}$ from the ambient space $\mathbb{R}^n$. Furthermore, the orthonormal basis functions of the Hilbert space $L^2(\mathcal{N}, \bar{q}^{-1})$ are the eigenfunctions of the adjoint (with respect to $L^2(\mathcal{N})$) of the operator, $\mathcal{L} = \nabla \log(\bar{q}) \cdot \nabla + \Delta$, that is constructed by the VBDM algorithm. Incidentally, the adjoint operator $\mathcal{L}^*$ is the Fokker-Planck operator of a gradient system forced by stochastic noises. The point is that this adjoint operator takes density functions of the weighted Hilbert space $L^2(\mathcal{N}, \bar{q}^{-1})$. Since the Hilbert space $L^2\left(\mathcal{N}, \bar{q}^{-1}\right)$ is a function space of some Fokker-Planck operator that acts on densities defined with respect to the geometry of data, then representing the conditional density with basis functions of the weighted Hilbert space $L^2\left(\mathcal{N}, \bar{q}^{-1}\right)$ is a natural choice. Thus, the error estimation in Theorem 2 is valid in controlling the error of the estimate.

Next, we will show that the representation of the true density p_{EX} in the ambient space $\mathbb{R}^3$ is not a function of $\mathcal{H}_{q^{-1}}(\mathcal{R}) \subset L^2\left(\mathcal{R}, q^{-1}\right)$, where for Hermite representation $\mathcal{R}$ is $\mathbb{R}^3$ and q is a normal distribution, and for Cosine representation $\mathcal{R}$ is a hyperrectangle containing the torus and q is a uniform distribution. Recall that the torus is parametrized by:

$$\mathbf{x} \equiv (x, y, z) = ((2 + r\sin(\theta))\cos(\phi), (2 + r\sin(\theta))\sin(\phi), r\cos(\theta)), \tag{A3}$$

where θ, ϕ are angles which make a full circle, and r is the radius of the tube known as the minor radius. All observation data are located on the torus with $r = 1$. Then, the generalized conditional density $p(r, \theta, \phi | D)$ in (r, θ, ϕ) coordinate can be defined using the Dirac delta function as follows,

$$p(r, \theta, \phi | D) = p_{\mathrm{IC}}(\theta, \phi | D)\,\delta(r - 1), \tag{A4}$$

where p_{IC}, defined in (A1), denotes the conditional density function in the intrinsic coordinate, (θ, ϕ), and δ is the Dirac delta function. After coordinate transformation, the density $p_{\mathrm{EX}} : \mathbb{R}^3 \to R$ can be obtained as

$$p_{\mathrm{EX}}(\mathbf{x} | D) = \frac{p(r, \theta, \phi | D)}{J} = \frac{p_{\mathrm{IC}}(\theta, \phi | D)\,\delta(r - 1)}{J}, \tag{A5}$$

where J is the Jacobian determinant $\det\left[\partial(x, y, z) / \partial(r, \theta, \phi)\right]$. It can be examined that $p_{\mathrm{EX}}(\mathbf{x} | D)$ is a generalized conditional density, that is, $\int_{\mathbb{R}^3} p_{\mathrm{EX}}(\mathbf{x} | D)d\mathbf{x} = 1$. Now it can be clearly seen that due to the Dirac delta function $\delta(r - 1)$ in (A5), the density $p_{\mathrm{EX}}(\mathbf{x} | D)$ is no longer in the weighted Hilbert space, $p_{\mathrm{EX}}(\mathbf{x} | D) \notin \mathcal{H}_{q^{-1}}(\mathcal{R})$. Consequently, the error estimation in Theorem 1 becomes invalid in controlling the error of the conditional density.

Here, the key point is that for Cosine and Hermite representations, the volume integral is with respect to $d\mathbf{x}$. The complete basis functions are obtained from tensor product of three sets of basis functions in (x, y, z) coordinates. In order to represent a conditional density function $p_{\mathrm{EX}}(\mathbf{x} | D)$ defined only on an intrinsically 2D torus domain, theoretically infinite number of basis functions are needed. However, numerically only finite number of basis functions can be used. Then, the density $p_{\mathrm{EX}}(\mathbf{x} | D)$ in (A5) cannot be well approximated for Hermite and Cosine representations (10). Moreover, if only finite number of Hermite or Cosine basis functions are used for representations, typically Gibbs phenomenon can be observed, i.e., the Dirac delta function $\delta(r - 1)$ in (A5) will be approximated by a function having a single tall spike at $r = 1$ with some oscillations at two sides along the r direction. On the other hand, the data-driven basis functions obtained via the diffusion maps algorithm are smooth

functions defined on the data manifold $\mathcal{N}$. Therefore, while the Gibbs phenomenon still occurs in this spectral expansion, it is due to finite truncation in representing a positive smooth functions (densities) on the data manifold, and not due to the singularity that occurs in the ambient direction as in (A5).

References

1. Kaipio, J.; Somersalo, E. *Statistical and Computational Inverse Problems*; Springer: New York, NY, USA, 2005.
2. Sullivan, T.J. *Introduction to Uncertainty Quantification*; Springer: Cham, Switzerland, 2015; Volume 63.
3. Dashti, M.; Stuart, A.M. The Bayesian Approach to Inverse Problems. In *Handbook of Uncertainty Quantification*; Ghanem, R., Higdon, D., Owhadi, H., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 311–428.
4. Brooks, S.; Gelman, A.; Jones, G.; Meng, X.L. *Handbook of Markov Chain Monte Carlo*; CRC press: London, UK, 2011.
5. Golightly, A.; Wilkinson, D. Markov chain Monte Carlo algorithms for SDE parameter estimation. **2010**, 253–276. Available online: http://www.mas.ncl.ac.uk/ nag48/diffchap.pdf (accessed on 2 June 2019).
6. Tavaré, S.; Balding, D.J.; Griffiths, R.C.; Donnelly, P. Inferring coalescence times from DNA sequence data. *Genetics* **1997**, *145*, 505–518. [PubMed]
7. Turner, B.M.; Van Zandt, T. A tutorial on approximate Bayesian computation. *J. Math. Psychol.* **2012**, *56*, 69–85. [CrossRef]
8. Neal, R.M. MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*; Brooks, S., Gelman, A., Jones, G., Meng, X.-L., Eds.; CRC Press: London, UK, 2011, Chapter 5, pp. 113–167.
9. Beck, J.L.; Au, S.K. Bayesian updating of structural models and reliability using Markov Chain Monte Carlo simulation. *J. Eng. Mech.* **2002**, *128*, 380–391. [CrossRef]
10. Haario, H.; Laine, M.; Mira, A.; Saksman, E. DRAM: Efficient adaptive MCMC. *Stat. Comput.* **2006**, *16*, 339–354. [CrossRef]
11. Higdon, D.; Kennedy, M.; Cavendish, J.C.; Cafeo, J.A.; Ryne, R.D. Combining field data and computer simulations for calibration and prediction. *SIAM J. Sci. Comput.* **2004**, *26*, 448–466. [CrossRef]
12. Marzouk, Y.; Najm, H.; Rahn, L. Stochastic spectral methods for efficient Bayesian solution of inverse problems. *J. Comput. Phys.* **2007**, *224*, 560–586. [CrossRef]
13. Marzouk, Y.; Xiu, D. A stochastic collocation approach to Bayesian inference in inverse problems. *Commun. Comput. Phys.* **2009**, *6*, 826–847. [CrossRef]
14. Huttunen, J.M.; Kaipio, J.P.; Somersalo, E. Approximation errors in nonstationary inverse problems. *Inverse Probl. Imag.* **2007**, *1*, 77–93. [CrossRef]
15. Nagel, J.B.; Sudret, B. Spectral likelihood expansions for Bayesian inference. *J. Comput. Phys.* **2016**, *309*, 267–294. [CrossRef]
16. Song, L.; Huang, J.; Smola, A.; Fukumizu, K. Hilbert space embeddings of conditional distributions with applications to dynamical systems. In Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, Quebec, Canada, 14–18 June 2009; ACM: New York, NY, USA, 2009; pp. 961–968.
17. Song, L.; Fukumizu, K.; Gretton, A. Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Process. Mag.* **2013**, *30*, 98–111. [CrossRef]
18. Berry, T.; Harlim, J. Correcting biased observation model error in data assimilation. *Mon. Weather Rev.* **2017**, *145*, 2833–2853. [CrossRef]
19. Coifman, R.R.; Lafon, S. Diffusion maps. *Appl. Comput. Harmon. Anal.* **2006**, *21*, 5–30. [CrossRef]
20. Berry, T.; Harlim, J. Variable bandwidth diffusion kernels. *Appl. Comput. Harmon. Anal.* **2016**, *40*, 68–96. [CrossRef]
21. Steinwart, I.; Christmann, A. *Support Vector Machines*; Springer: New York, NY, USA, 2008.
22. Berry, T.; Harlim, J. Forecasting turbulent modes with nonparametric diffusion models: Learning from noisy data. *Physica D* **2016**, *320*, 57–76. [CrossRef]
23. Harlim, J. *Data-Driven Computational Methods: Parameter and Operator Estimations*; Cambridge University Press: Cambridge, UK, 2018.

24. Berry, T.; Sauer, T. Consistent manifold representation for topological data analysis. *Found. Data Sci.* **2019**, *1*, 1–38. [CrossRef]

25. Nyström, E.J. Über die praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben. *Acta Math.* **1930**, *54*, 185–204. (In German) [CrossRef]

26. Aronszajn, N. Theory of reproducing kernels. *Trans. Am. Math. Soc.* **1950**, *68*, 337–404. [CrossRef]

27. Harlim, J.; Yang, H. Diffusion Forecasting Model with Basis Functions from QR-Decomposition. *J. Nonlinear Sci.* **2018**, *28*, 847–872. [CrossRef]

28. Lorenz, E. Predictability: A problem partly solved. In Proceedings of the Seminar on Predictability, Shinfield Park, Reading, UK, 4–8 September 1995; ECMWF: Reading, UK, 1995; Volume 1, pp. 1–18.

29. Majda, A.; Abramov, R.; Grote, M. *Information Theory and Stochastics for Multiscale Nonlinear Systems*; (CRM Monograph Series); American Mathematical Society: Providence, RI, USA, 2005.

30. Harlim, J.; Li, X.; Zhang, H. A Parameter Estimation Method Using Linear Response Statistics. *J. Stat. Phys.* **2017**, *168*, 146–170. [CrossRef]

31. Zhang, H.; Li, X.; Harlim, J. A Parameter Estimation Method Using Linear Response Statistics: Numerical Scheme. *Chaos* **2019**, *29*, 033101. [CrossRef]

32. Abramov, R.; Majda, A. Blended response algorithm for linear fluctuation-dissipation for complex nonlinear dynamical systems. *Nonlinearity* **2007**, *20*, 2793–2821. [CrossRef]

33. Loftsgaarden, D.O.; Quesenberry, C.P. A nonparametric estimate of a multivariate density function. *Ann. Math. Stat.* **1965**, pp. 1049–1051. [CrossRef]

34. Mack, Y.; Rosenblatt, M. Multivariate k-nearest neighbor density estimates. *J. Multivar. Anal.* **1979**, *9*, 1–15. [CrossRef]