Estimating Parameters in Complex Systems with Functional Outputs—A Wavelet-based Approximate Bayesian Computation Approach

Hongxiao Zhu*, Ruijin Lu*, Chen Ming, Anupam K. Gupta and Rolf Müller

Abstract We consider a family of parameter estimation problems involving functional data. In these problems, the relationship between functional data and the underlying parameters cannot be explicitly specified using a likelihood function. These situations often occur when functional data arises from a complex system and only numerical simulations (through a simulator) can be used to describe the underlying data-generating mechanism. To estimate the unknown parameters under these scenarios, we introduce a wavelet-based approximate Bayesian computation (wABC) approach that is likelihood-free and computationally scalable to functional data measured on a dense, high-dimensional grid. The proposed approach relies on near-lossless wavelet decomposition and compression to reduce the high-correlation between measurement points and the high-dimensionality. We adopt a Markov chain Monte Carlo algorithm with a Metropolis-Hastings sampler to obtain posterior samples of the parameters for Bayesian inference. To avoid expensive simulations from the simulator in the approximate Bayesian computation, a Gaussian process surrogate for the simulator is introduced, and the uncertainty of the resulting sampler is controlled by calculating the expected error rate of the acceptance probability. We motivate our approach and demonstrate its performance using the foliage-echo data generated by a sonar simulation system. Our Bayesian posterior inference provides the joint posterior distribution of all underlying parameters, which is otherwise intractable using existing analytical methods.

*These authors contributed equally to this work.
Hongxiao Zhu and Ruijin Lu
Department of Statistics, Virginia Tech, 250 Drillfield Drive, MC0439, Blacksburg, VA 24061 USA e-mail: hongxiao@vt.edu, lruijin@vt.edu

Chen Ming, Anupam K. Gupta and Rolf Müller
Department of Mechanical Engineering, Virginia Tech, 1075 Life Science Cir, MC0917, Blacksburg, VA 24061 USA e-mail: cming@vt.edu, anupamkg@vt.edu, rolf.mueller@vt.edu

1 Introduction

Functional data, such as signals, surfaces, and images, are frequently encountered in many scientific disciplines. The increased prevalence of such data promotes the development of *functional data analysis* [20, 7, 11, 29]. While considerable efforts have been made to the preprocessing [19, 25], estimation [21, 34, 33], and regression analysis [5, 4, 36, 24, 16] of functional data, existing approaches primarily rely on linking functional observations with the unknown parameters via a likelihood or an objective function. Many applications, however, involve inferring parameters when such linkage is implicit or difficult to specify. In this paper, we consider a family of parameter estimation problems under such situations.

Figure 1 provides a conceptual demonstration of the estimation problems we consider. The black box represents an unknown complex system that takes the parameter θ as input and produces functional observations $\{Y_i(t)\}$ as outputs. Our goal is to estimate the underlying parameter θ based on the observed functional outputs. If the relationship between $\{Y_i(t)\}$ and θ is known, for example, if $\{Y_i(t)\}$

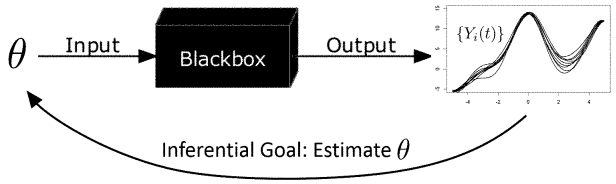

Fig. 1 A conceptual demonstration of the parameter estimation problems we consider.

are independent and identically distributed Gaussian processes with mean zero and a covariance kernel that depends on θ, we can estimate θ through maximum likelihood or Bayesian method. There are, however, many other situations in which the true linkage between $\{Y_i(t)\}$ and θ is more complicated, and scientists use physical rules and/or mathematical equations to model such linkage. To illustrate these situations, we provide two examples—the light detection and ranging (LIDAR) data and the foliage-echo data.

1. *The LIDAR data.* LIDAR is an optical remote-sensing technique that uses laser light to measure targets and produces high-resolution functional data. For example, authors in [32] considered LIDAR data measured on an aerosol cloud. During the measurement, a point source laser was transmitted into an aerosol cloud at multiple wavelengths and over multiple time points. The laser light was then scattered by the aerosol cloud and reflected back to a receiver. The resulting data can be modeled by $Y(t,z) = g(t,z) + \varepsilon(t,z)$, where t is time, z is the range value, $Y(t,z)$ is the random surface that can be observed, $g(t,z)$ is the underlying true signal, and $\varepsilon(t,z)$ is the random measurement error. The linkage between $g(t,z)$ and the parameters of interest is implicit, described by a partial differential

equation (PDE):

$$\frac{\partial g(t,z)}{\partial t} - \theta_D \frac{\partial^2 g(t,z)}{\partial z^2} - \theta_S \frac{\partial g(t,z)}{\partial z} - \theta_A g(t,z) = 0,$$

subject to boundary conditions. Here, the parameters θ_D, θ_S, and θ_A denote the diffusion rate, the drift shift, and the reaction rate respectively, which reflect the physical properties of the laser light reflection. Consequently, the relationship between the functional observation $Y(t,z)$ and the parameters is implicit, and one cannot write the likelihood of $Y(t,z)$ in terms of the parameters explicitly.

2. *The foliage-echo data*. The foliage-echo data represents a more general situation when functional data is produced by a complicated system which cannot be described using a single formula (e.g., a PDE). During the measurement, an active sonar system transmits acoustic waves into tree foliages, and the waves reflected back from the foliages (i.e. the echoes) are received. While the mechanism of sound propagation and reflection is complicated, we are able to simulate echoes using a simulator by applying acoustic laws under simplified assumptions. Details of the simulation are described in Section 2 and the Appendix. Our goal is to estimate properties of the foliages, such as the density of the leaves (i.e., how many leaves per cubic meter), based on the echoes.

The above two examples demonstrate functional data produced by complex systems. These systems have the following characteristics: (1) Due to the complexity of the underlying physical rules, the parameter estimation is a difficult inverse problem which may be ill-posed, meaning that the solution to the parameter estimation may not be unique. For example, both LIDAR and foliage-echo examples are remote sensing problems in which the data are aggregations of reflected waveforms from numerous reflectors; therefore, it is possible that different combinations of the model parameters result in the same/similar data outputs. Furthermore, analytical or numerical solution to these inverse problems is often hard to find. (2) One can numerically simulate data from a physical/mathematical model (e.g., a PDE or a more complicated simulator), but the simulation may be computationally intensive. (3) The data-generation procedure of the complex system involves random variables, hence, it produces random functional outputs for a given set of parameters. For example, in both LIDAR and foliage-echo examples, randomness may be caused by measurement error and/or numerous reflecting facets whose size, location, and orientation follow certain probability distributions. (4) It is often difficult to explicitly link the functional outputs with the underlying parameters via a likelihood or an objective function. (5) The functional outputs are often measured on a dense, high-dimensional grid.

For systems that can be described using ordinary differential equations (ODEs) or PDEs, such as the LIDAR data case, estimation approaches based on regularized optimization, also called *parameter cascading*, have been proposed [18, 14, 32, 35]. These methods, however, are not suitable for systems that cannot be described by ODEs or PDEs. In this paper, we propose a wavelet-based approximate Bayesian computation (wABC) approach that is applicable to general complex systems—

systems that include ODE and PDE as special cases. For this reason, we will use the more general foliage-echo data as our primary example.

The proposed wABC approach inherits the "likelihood-free" property of the traditional approximate Bayesian computation (ABC) [15, 26] through bypassing analytical evaluations of the likelihood function. The bypassing is achieved through approximating the likelihood function evaluation by simulation. The basic idea is illustrated in Figure 2. Specifically, instead of evaluating the likelihood, the ABC approach first samples a candidate parameter θ^* from the prior distribution $\pi(\theta)$, then simulates data $\{X_i\}$ from a "simulator" of the system by treating θ^* as the input. If the simulated data is "close to" the observed data, the candidate parameter θ^* is accepted, otherwise it is rejected. A more detailed review of ABC can be found in Section 3.1.

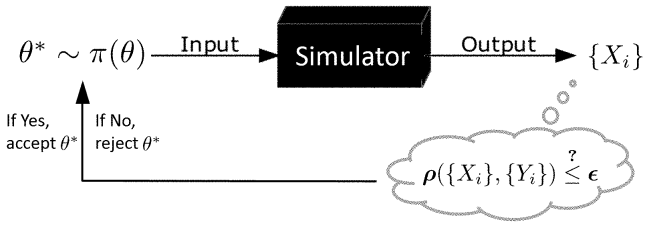

Fig. 2 The basic idea of the ABC method. Here, $\{X_i\}$ represent the simulated data, $\{Y_i\}$ represent the observed data, and $\rho(\cdot,\cdot)$ measures how "close" the simulated data are to the observed data.

Despite their flexibility in handling complex systems, as a simulation-based approach, the ABC method suffers from low efficiency when the dimension of the observed data increases and when the "simulator" becomes computationally expensive. As the dimension of the data increases, the criterion $\rho(\{X_i(t)\},\{Y_i(t)\}) \leq \varepsilon$ is harder to be satisfied, resulting in lower acceptance rate. When the "simulator" becomes moderately expensive, even on the scale of a few seconds per simulation, accepting 1000 samples of θ would require hours of calculation, and the computation quickly becomes intractable when the acceptance rate drops. Our proposed wABC approach extends beyond existing ABC by allowing functional outputs measured on high-dimensional grid, yet still remains computationally tractable. It relies on the near-lossless wavelet decomposition and compression to reduce the high-correlation between measurement points and the high-dimensionality, and adopts a Markov chain Monte Carlo algorithm with a Metropolis-Hastings sampler to obtain posterior samples of the parameters. To avoid expensive simulations, a Gaussian process surrogate for the simulator is introduced, and the uncertainty of the resulting sampler is controlled by calculating the expected error rate of the acceptance probability.

To our knowledge, the proposed wABC approach is the first that estimates parameters in complex systems based on functional outputs measured on a dense, high-dimensional grid. It is generally applicable to various physical, chemical, and biological systems that facilitate numerical simulations. Compared with existing

functional data analytical tools, our approach has the following advantages: (1) It is likelihood-free. It takes full advantages of the physical/mathematical rules that connect data with the parameter. (2) It can characterize various linear or nonlinear data-parameter relationships. (3) It produces the joint posterior distribution of the parameters with various multi-modality and shape structures. (4) It is scalable to functional outputs measured on high-dimensional grids as well as expensive simulations. Our results for the simulated foliage-echo data demonstrate the effectiveness of the proposed method in estimating parameters.

2 A Motivating Example: The Foliage-echo Simulation System

While the method we propose is generally applicable to various complex systems, it is initially motivated by the foliage-echo study. The goal of the study is to estimate the statistical properties of tree foliages, i.e., the density of the leaves, the average size of the leaves, and the average orientation of the leaves, based on the echo signals captured by a sonar device.

Figure 3 shows the working mechanism of an active sonar, which consists of an emitter that ensonifies the environment and a receiver that records the returning echoes. The transmitter emits acoustic waves and the receiver collects echoes reflected from objects in the environment. The echo signals carry information about the targets, hence have been used for various identification and navigation tasks [27]. In natural environments, an echo signal is the superposition of reflected waveforms from numerous scatterers, e.g., foliage leaves, rocks in uneven natural terrains, thus is highly stochastic.

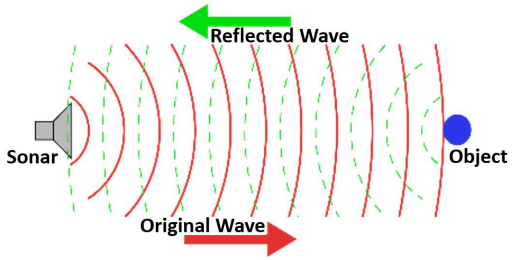

Fig. 3 The principle of an active sonar. This figure was created based on an online figure available at the Wikipedia website on Sonar [31] (https://en.wikipedia.org/wiki/Sonar).

To study the foliage echoes, we establish a computational model to simulate a natural sonar scene in a three-dimensional (3-d) space. The scene is demonstrated in Figure 4 (a), which consists of an active sonar sensor and a cluster of tree leaves. The sensor is located at the origin. It emits ultrasonic waves towards the positive x-axis direction. The tree foliages are uniformly located in a $[1,10]\times[-2,2]\times[-2,2]$ region in 3-d. The total number of leaves is determined by the leaf density—the

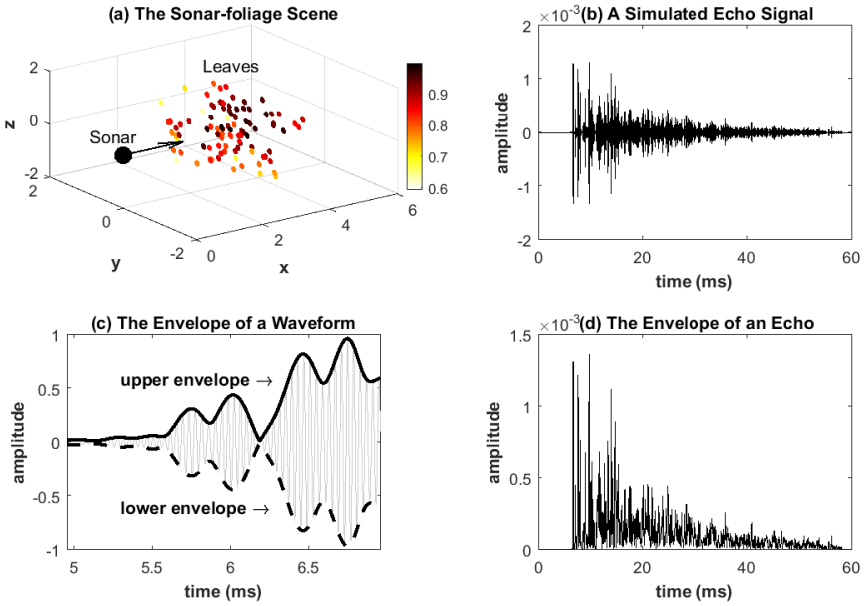

Fig. 4 The Foliage-echo Simulation. (a) The sonar scene in 3-d. The color indicates the sound intensity leaves receive/reflect (scaled to $[0,1]$). (b) A simulated echo signal with leaf density of 30 (in number of leaves per cubic meter), leaf radius of 0.0171 (in meter), and leaf orientation of 45 (in degree). (c) A demonstration of the upper and lower envelopes of the waveform. (d) The echo envelope extracted from the echo signal in (b).

number of leaves per cubic meter, denoted by θ_1. The leaf shapes are approximated by planar circular disks with radius (denoted by a) randomly sampled from a normal distribution $N(\theta_2, 0.1\theta_2)$, where θ_2 denotes the mean radius. The orientation of each leaf relative to the sonar is determined by two angles: (1) the angle between the leaf normal vector and pulse direction (the positive x-axis direction), which follows a truncated normal distribution $N(\beta|\theta_3, 5)1_{\{0<x<90\}}$ with θ_3 the mean angle and 5 the variance; and (2) the angle that describes the rotation of the leaf normal vector around the pulse direction clockwisely, which follows a uniform distribution in the range of $[0, 2\pi)$. Based on these two angles, we further calculate the incident angle—the angle between the leaf's normal direction and the sonar-leaf center line. We denote the incident angle by β. With these setups and the specification of the acoustic properties of the sonar, echoes are simulated following acoustic laws of sound emission, propagation, and reflection [3]. More technical details of the simulator are described in the Appendix.

The above simulation model constitutes a physical system with three inputs: the leaf density (θ_1), the mean leaf radius (θ_2), and the mean leaf orientation (θ_3). The output is an echo signal as demonstrated in Figure 4 (b). The output echo signal is a temporal waveform measured from 0 to 60 milliseconds with a sampling rate of 400

kHz. The total number of measurement points is $24,000$ for each echo. The parameters $(\theta_1, \theta_2, \theta_3)$ summarize the statistical properties of the foliage targets. Therefore, estimating these parameters based on the echo signals provides us knowledge of the targets. While the current study only involves echoes simulated from a physical model, our ultimate hope is to use the proposed estimation approach to infer target properties based on echoes collected in a real scene.

Directly modeling the echo signals is difficult because the echoes contain information about both emitted signals and the target properties. Since sound reflection from stationary targets does not change the carrier frequency of the emitted signal, information about targets is contained in the amplitude modulation, which is captured by the *envelopes* of the echo signals. We therefore perform a preprocessing step to extract the echo envelopes, and use this data for our analysis. The envelope of a signal is the boundary curve within which all amplitude values of the signal are contained. A conceptual demonstration is shown in Figure 4 (c). The envelope of an echo retains the target-specific information by capturing the low frequency amplitude variations, which makes it an ideal representation of echo signals. In the sonar echo data, since the upper and the lower envelopes are always symmetric, we only consider the upper envelopes in our data analysis. The envelope signal extracted from the echo in Figure 4 (b) is shown in Figure 4 (d).

3 Wavelet-based Approximate Bayesian Computation

The foliage-echo data example demonstrated in Section 2 represents a family of parameter estimation problems involving functional data. In these problems, functional data is related to the parameters of interest through a complex system guided by physical or mathematical rules. As a result, one cannot explicitly write the likelihood of the functional outputs as a function of the parameters. To facilitate parameter estimation under these scenarios, we propose a wavelet-based Approximate Bayesian Computation (wABC) approach. The logic behind the main concepts introduced in Sections 3.1–3.4 and their connections with wABC are illustrated in Figure 5. Section 3.1 reviews the general ABC approach, which is the foundation for the proposed wABC approach. In order to facilitate functional outputs measured on a dense, high-dimensional grid, we represent functional data through wavelet basis expansion and perform a wavelet compression to reduce dimension; details are in Section 3.2. For simulators that are computationally expensive, we further introduce the Gaussian process surrogate for the simulator to enable fast simulation; this is discussed in Section 3.3. Sections 3.1–3.3 constitute the general framework of wABC. Finally, in Section 3.4, we introduce a method to control the uncertainty of the decision-making in wABC.

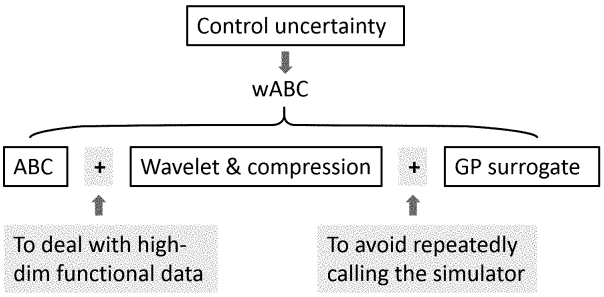

Fig. 5 The logic behind the concepts introduced in Sections 3.1–3.4 and their connections with wABC.

3.1 Review of Approximate Bayesian Computation

Let Y denote a random element whose realizations are the observed data and let θ denote a parameter that determines the distribution of Y. In a typical Bayesian setup, one computes the posterior distribution $\pi(\theta|Y) \propto \pi(Y|\theta)\pi(\theta)$, where $\pi(Y|\theta)$ is the likelihood that relates Y to the parameter θ and $\pi(\theta)$ is the prior distribution for θ. Approximate Bayesian Computation (ABC), initially proposed by [17], aims to approximate the posterior distribution $\pi(\theta|Y)$ without explicitly specifying the likelihood $\pi(Y|\theta)$. In particular, we assume that $\pi(Y|\theta)$ is unknown, but there is a simulation model, often denoted by $\pi(X|\theta)$, that produces simulated data X given θ^*. We sometimes call X the pseudo-data. Here, θ^* is an arbitrary sample from the prior distribution $\pi(\theta)$. If X is "close to" Y, we retain θ^* as a sample of $\pi(\theta|Y)$, otherwise, we reject θ^* and repeat the procedure with a new θ^*. This procedure, as illustrated in Figure 2, will be repeated until the desired amount of "good samples" is collected. In ABC, we often use a distance measure $\rho(\cdot,\cdot)$ to determine how close X is to Y. For example, in the univariate case, by letting $\rho(X,Y) = |X-Y|$, we will retain θ^* when $|X-Y| \leq \varepsilon$ for a small ε.

The above procedure indeed produces samples for the distribution $\pi(\theta|\{\rho(X,Y) \leq \varepsilon\})$, a distribution that is identical to $\pi(\theta|Y)$ when $\varepsilon = 0$ (i.e., $X = Y$). However, since $\{\text{X=Y}\}$ happens with probability 0 for continuous random variables, in practice, we can only require $\rho(X,Y) \leq \varepsilon$ for a small discrepancy ε, which results in $\pi(\theta|\{\rho(X,Y) \leq \varepsilon\})$. The distribution $\pi(\theta|\{\rho(X,Y) \leq \varepsilon\})$ serves as an approximation of $\pi(\theta|Y)$ when ε is small, i.e.,

$$\pi(\theta|Y) \approx \pi(\theta|\rho(X,Y) \leq \varepsilon), \text{ for a small } \varepsilon.$$

When multiple samples are observed, we index the data by Y_i, $i = 1,\ldots,n$ and denote $\mathbf{Y} = \{Y_1,\ldots,Y_n\}$. In this case, ABC can be performed by sampling $\mathbf{X} = \{X_1,\ldots,X_m\}$ based on each θ^*, and define $\rho(\cdot,\cdot)$ based on a summary statistic $S(\cdot)$ of the samples. If $S(\mathbf{Y})$ is a sufficient statistic for θ, then $S(\mathbf{Y})$ contains all information about θ, therefore $\pi(\theta|\mathbf{Y}) = \pi(\theta|S(\mathbf{Y}))$, which can be shown by applying the Fisher-Neyman factorization theorem [13]. The right-hand side of the equation

$\pi(\theta|S(\mathbf{Y}))$ can be further approximated by $\pi(\theta|\rho(S(\mathbf{X}),S(\mathbf{Y}))\leq\varepsilon)$ using the ABC. For example, if $\{Y_1,\ldots,Y_n\}$ is a random sample from a Bernoulli distribution with mean θ, then one can define $\rho(\mathbf{X},\mathbf{Y})=|\bar{Y}-\bar{X}|$, where $S(\mathbf{Y})=\bar{Y}$ is the sample mean, a sufficient statistic for θ.

Markov Chain Monte Carlo for ABC. The traditional ABC procedure relies on accepting θ^* when $\rho(S(\mathbf{X}),S(\mathbf{Y}))\leq\varepsilon$. This procedure can be embarrassingly inefficient because of two reasons: (i) A good sufficient statistic can be hard to find. Sometimes one has to use the original data set as the sufficient statistic. (ii) The acceptance rate can be extremely low especially when the statistic $S(\cdot)$ or the parameter θ is of high dimension. Various alternative algorithms have been proposed to improve the computational efficiency of ABC. Here, we review an Markov chain Monte Carlo (MCMC) algorithm using the Metropolis-Hastings (MH) sampler. More discussions of the MCMC algorithm for ABC can be found in [30, 6, 1, 22], among others. First, we transfer the acceptance criterion $\rho(S(\mathbf{X}),S(\mathbf{Y}))\leq\varepsilon$ to a probability density function $\pi_\varepsilon(S(\mathbf{Y})\mid S(\mathbf{X}))$ controlled by the discrepancy parameter ε. For example, with an independent Gaussian assumption, we may write

$$\pi_\varepsilon(S(\mathbf{Y})\mid S(\mathbf{X}))=(2\pi\varepsilon)^{-J/2}\exp\{-\frac{1}{2\varepsilon^2}(S(\mathbf{X})-S(\mathbf{Y}))^T(S(\mathbf{X})-S(\mathbf{Y}))\},\quad(1)$$

where J is the dimension of the sufficient statistic $S(\cdot)$. With this representation, we can approximate the likelihood $\pi(S(\mathbf{Y})\mid\theta)$ by $\pi_\varepsilon(S(\mathbf{Y})\mid\theta)$, and the latter can be approximated using the Monte Carlo integration

$$\pi_\varepsilon(S(\mathbf{Y})\mid\theta)=\int\pi_\varepsilon(S(\mathbf{Y})\mid S(\mathbf{X}))\pi(S(\mathbf{X})\mid\theta)\,dS(\mathbf{X})\approx\frac{1}{H}\sum_{g=1}^{H}\pi_\varepsilon(S(\mathbf{Y})\mid S(\mathbf{X}^{(g)})).\quad(2)$$

Here, $\{\mathbf{X}^{(g)},g=1,\ldots,H\}$ denote H samples of the pseudo-data generated from the simulator, $\pi(\mathbf{X}\mid\theta)$. Note that we do not need to evaluate $\pi(S(\mathbf{X})\mid\theta)$ in equation (2). We just need to sample from it. Based on the approximated likelihood, we can design a MCMC algorithm by assuming an proposal distribution $q(\theta^*|\theta)$. We accept the proposed θ^* with probability

$$\alpha(\theta^*|\theta)=\min\left\{1,\frac{\pi(\theta^*)\pi_\varepsilon(S(\mathbf{Y})\mid\theta^*)q(\theta|\theta^*)}{\pi(\theta)\pi_\varepsilon(S(\mathbf{Y})\mid\theta)q(\theta^*|\theta)}\right\}.$$

The above MCMC algorithm provides improved mixing for the posterior samples than the traditional rejection-based ABC algorithm. However, it requires H repeated calls to the simulator in order to compute the approximation in equation (2), and this has to be performed during each MCMC iteration. Here, H needs to be large enough to guarantee a good approximation, e.g., $H=1000$ is reasonable if $\pi(S(\mathbf{X})\mid\theta)$ is a Gamma distribution. Repeated sampling can be a computational burden when the simulator runs slow. In Section 3.3, we adopt a Gaussian process surrogate (GPS) for the simulator following the idea of [1], which substantially reduces the number of simulation calls.

3.2 Wavelet Representation and Compression of Functional Data

While the idea of ABC is straightforward to follow, it can be inefficient due to a number of assumptions and approximations that may not be easily satisfied. One assumption is the existence of a sufficient statistic for the parameters of interest. Given a random sample $\mathbf{Y} = \{Y_1, \ldots, Y_n\}$, the determination of a sufficient statistic $S(\mathbf{Y})$ for θ is often difficult without knowing the distribution of Y_i. Although one can always choose the data itself as the sufficient statistic, doing so only makes the specification of the distance measure $\rho(\cdot,\cdot)$ extremely difficult (because the dimension of $\mathbf{Y}$ is high). This issue is particularly severe for high dimensional vectors and functional data. In our foliage-echo example, an echo envelope is of dimension $24,000$, therefore, the data $\mathbf{Y}$ can be written as a n-by-$24,000$ matrix. Given that the relationship between the data and the parameters is implicit, determining a sufficient statistics for $(\theta_1, \theta_2, \theta_3)$ given $\mathbf{Y}$ is practically intractable.

To facilitate the efficient performance of ABC for functional data measured on a dense, high-dimensional grid, we adopt a strategy that achieves de-correlation and compression so that functional observations can be parsimoniously represented in a much lower dimensional setting. In particular, we represent the functional data by a multi-scale wavelet basis. Given a set of multi-scale wavelet basis functions $\{\psi_{jk}; j = 1, \ldots, J,\ k = 1, \ldots, K_j\}$ and a scale function (the father wavelet) $\{\psi_{0k}; k = 1, \ldots, K_0\}$, we can expand a functional observation $Y(t)$ by $Y(t) = \sum_{j=0}^{J} \sum_{k=1}^{K_j} d_{jk} \psi_{jk}(t)$. Here, d_{jk} is the wavelet coefficient at scale j and location k. For functional data measured on an equally spaced grid, this representation is *lossless*, i.e., providing an exact representation of the original data. Therefore, $\{d_{jk}\}$ contain the same amount of information as $Y(t)$ thus can be treated as a sufficient statistic for θ. We can denote the sufficient statistics of $\mathbf{Y}$ as $S(\mathbf{Y}) = \mathbf{D}$, where $\mathbf{D} = (d_{ijk})$ is a n-by-K matrix and $K = \sum_{j=0}^{J} K_j$. In general, the wavelet transformation is not the only option. It is possible to construct lossless transforms with other basis functions (e.g. Spline or Fourier bases), or construct an approximately lossless transformation with a basis $\{B_k(t), k = 1, \ldots, K\}$ that satisfies $|Y(t) - \sum_{k=1}^{K} d_k B_k(t)| < \delta$ for all t and a small δ.

The wavelet representation has two advantages: the coefficients $\{d_{jk}\}$ are sparse, meaning that most coefficients are zero or close-to-zero, and they are approximately uncorrelated. These properties bring two types of convenience to the specification of the distance measure in ABC. First, since components in $\{d_{jk}\}$ are approximately uncorrelated, the conditional distribution $\pi_\varepsilon(S(\mathbf{Y}) \mid S(\mathbf{X}))$ can be specified following equation (1), i.e., assuming that components of $S(\mathbf{Y})$ (or $S(\mathbf{X})$) are mutually independent of each other. Second, the sparsity of the wavelet coefficients makes the wavelet compression feasible.

Wavelet Compression. For many high-dimensional problems, representing the data in a much lower dimensional space brings tremendous convenience to data storage and processing. This is also true in the ABC context. Let $\mathbf{D} = (d_{ijk})$ denote the n by K matrix of wavelet coefficients, and the ith row corresponds to the wavelet coefficients of the ith functional observation. Since $\mathbf{D}$ is sparse, many components of $\mathbf{D}$

are zero or close-to-zero, therefore does not contain essential information about the parameter. Wavelet compression removes zero or close-to-zero components while retaining the large components. The compressed matrix, denoted by $\widetilde{\mathbf{D}}$, is nearly lossless, thus can be used as an approximately sufficient statistic for θ. To compress $\mathbf{D}$, we retain K_1 columns of $\mathbf{D}$ so that the proportion of energy retained is greater than (or equal to) a threshold δ_1 (e.g., $\delta_1 = 0.999$) for each function. Here, the proportion of energy retained for a function $Y_i(t)$ is defined by $\sum_{(j,k)\in C_1} d_{ijk}^2 / \sum_{(j,k)} d_{ijk}^2$, where C_1 is the set of scale and location indices that correspond to columns retained in $\widetilde{\mathbf{D}}$.

The wavelet representation and compression introduced above provide an effective way to transform the functional observation $\mathbf{Y}$ to wavelet coefficient matrix $\mathbf{D}$ in the wavelet domain, and to reduce the dimension of $\mathbf{D}$ from n-by-K to n-by-K_1. The compression also has the effect of removing high frequency noise in functional data. The reduced data $\widetilde{\mathbf{D}}$ will be treated as a sufficient statistics of $\mathbf{Y}$ to be used in the MCMC sampling scheme for wABC.

3.3 A Gaussian Process Surrogate for the Simulator

As discussed in Section 3.1, although the MCMC method can provide better mixing than the traditional rejection-based ABC method, it requires sampling from the simulator H times during each MCMC iteration. Even if each simulation only requires a moderate amount of time, running a large amount of MCMC iterations can be computationally intractable. For example, our foliage-echo simulator takes 2.3 seconds to simulate one echo envelope. If the MCMC algorithm has an acceptance rate of 30%, $H = 100$, and the number of independent samples in $\mathbf{X}$ is $m = 3$, the expected time needed to obtain 1000 posterior samples of θ is around 639 hours (26.6 days). It is possible to use parallel computing at the stage of computing $\pi_\varepsilon(S(\mathbf{Y}) \mid \theta)$, i.e., during each MCMC, the H samples of $\mathbf{X}$ (which contain Hm echoes) can be performed in parallel using a multi-core computing server. However, it may still take days to obtain 1000 posterior samples of θ because the number of computing cores one has access to is often limited. The modern graphics processing units (GPU) based computing system provides far more computing cores [23], but each core can only deal with relatively simple calculation, therefore may not be suitable for the large-scale matrix calculations required by our simulator. When the speed of the simulator cannot be improved any further, a good solution is to adopt a strategy that requires less calls of the simulator. We now introduce a GPS for the simulator following the idea of [1]. GPS can substantially reduce the number of simulation calls in the MCMC.

We explain the GPS in the context of the foliage-echo example. Suppose that J columns of $\mathbf{D}$ are retained after wavelet compression. Let $\widetilde{\mathbf{D}}_y = (\mathbf{d}_y^1, \ldots, \mathbf{d}_y^J)$ denote the n-by-J matrix of wavelet coefficients after compression, where each $\mathbf{d}_y^j$ is an n-by-1 vector. In the foliage-echo example, The randomness in the leaf location, orientation, and radius causes random fluctuations in the n samples. These

fluctuations reflect the leaf-specific information, i.e., exact locations, orientations, and radii of leaves in a scene, which is not relevant to the population parameters $(\theta_1, \theta_2, \theta_3)$. Therefore, we remove the random fluctuation by averaging each $\mathbf{d}_y^j$ across its n entries, resulting in a scalar $\bar{d}_y^j$. Denote the averaged wavelet coefficients by $\bar{\mathbf{D}}_y = (\bar{d}_y^1, \ldots, \bar{d}_y^J)^T$. We will use $S(\mathbf{Y}) = \bar{\mathbf{D}}_y$ in the analysis of foliage-sonar data. Since the wavelet coefficients in $\bar{\mathbf{D}}_y$ are approximately independent of each other, we will calculate the likelihood $\pi_\varepsilon(\bar{d}_y^j \mid \theta)$ for each j independently. We assume that

$$\bar{d}_y^j = \bar{d}_x^j + e^j, \quad e^j \sim N(0, \varepsilon^2). \tag{3}$$

Here, $\bar{d}_x^j$ is the jth averaged wavelet coefficients based on the simulated samples $\mathbf{X} = \{X_1, \ldots, X_m\}$. Model (3) is equivalent to assuming that $\pi_\varepsilon(\bar{d}_y^j \mid \bar{d}_x^j)$ corresponds to a $N(\bar{d}_x^j, \varepsilon^2)$ distribution. We further approximate the simulator distribution $\pi(\bar{d}_x^j \mid \theta)$ by assuming that $\bar{d}_x^j$ follows a Gaussian process (GP) regression model:

$$\bar{d}_x^j = f_j(\theta) + r_j, \quad f_j(\theta) \sim GP(0, k_j(\theta, \theta^*)), \quad r_j \sim N(0, \sigma_j^2), \tag{4}$$

where $f_j(\theta)$ is an unknown GP with mean zero and a pre-specified covariance kernel $k_j(\theta, \theta^*)$. For example, a commonly used covariance kernel is the squared exponential kernel $k_j(\theta, \theta^*) = \phi_j^2 \exp\{-||\theta - \theta^*||^2/(2\tau_j^2)\}$. Since both (3) and (4) induce Gaussian distributions, we can analytically calculate $\pi_\varepsilon(\bar{d}_y^j \mid \theta)$ by integrating out $\bar{d}_x^j$. This analytical integration avoids the need to perform approximation using Monte Carlo integration as described in equation (2). We call the GP regression model (4) a GPS. The main idea is to train a GP model on a grid of θ and use it to replace the simulation distribution $\pi_\varepsilon(\bar{d}_y^j \mid \theta)$. This strategy avoids the need of frequently calling the simulator during the MCMC iteration.

Specifically, we calculate $\pi_\varepsilon(\bar{d}_y^j \mid \theta)$ following a three-step procedure.

1. Produce a grid of values $\Theta = (\theta_1, \ldots, \theta_A)^T$ on the domain of θ, generate $\mathbf{X} = \{X_1, \ldots, X_m\}$ at each grid point, perform wavelet decomposition and compression of $\mathbf{X}$, and average the wavelets coefficients across the m samples. This results in a list of "input-output" pairs $\{(\theta_i, \bar{d}_{x,i}^j), i = 1, \ldots, A\}$, which will be treated as the training data for estimating the function $f_j(\theta)$.
2. Given a pair of values (θ^*, θ), we will calculate the GP predictive distribution on (θ^*, θ) using the conditional distribution, which gives $N(\mu^j_{(\theta^*,\theta)|\Theta}, \Sigma^j_{(\theta^*,\theta)|\Theta})$, where

$$\mu^j_{(\theta^*,\theta)|\Theta} = \begin{pmatrix} \mathbf{k}_{\theta^*,\Theta} \\ \mathbf{k}_{\theta,\Theta} \end{pmatrix} \left(\mathbf{K}_{\Theta,\Theta} + \sigma_j^2 \mathbf{I}\right)^{-1} \bar{\mathbf{d}}_x^j, \tag{5}$$

$$\Sigma^j_{(\theta^*,\theta)|\Theta} = \begin{pmatrix} k_{\theta^*,\theta^*} & k_{\theta^*,\theta} \\ k_{\theta,\theta^*} & k_{\theta,\theta} \end{pmatrix} - \begin{pmatrix} \mathbf{k}_{\theta^*,\Theta} \\ \mathbf{k}_{\theta,\Theta} \end{pmatrix} \left(\mathbf{K}_{\Theta,\Theta} + \sigma_j^2 \mathbf{I}\right)^{-1} \begin{pmatrix} \mathbf{k}_{\theta^*,\Theta} \\ \mathbf{k}_{\theta,\Theta} \end{pmatrix}^T. \tag{6}$$

Here, $\bar{\mathbf{d}}_x^j = (\bar{d}_{x,1}^j, \ldots, \bar{d}_{x,A}^j)^T$ is an A-by-1 vector of training points, $\mathbf{k}_{\theta^*,\Theta}$ is a 1-by-A vector consisting of kernel evaluations at θ^* and components in Θ, $\mathbf{K}_{\Theta,\Theta}$ is an A-by-A matrix consisting of kernel evaluations at two components in Θ, and

$k_{\theta^*,\theta} = k(\theta^*, \theta)$. We treat the above GP conditional distribution as a *surrogate* of the simulator. In Figure 6, we compared the prediction performance of the GPS at a test value of θ_1 with the sample estimate obtained from data directly sampled from the simulator. Here, we have fixed θ_2 and θ_3, treating θ_1 as the parameter to be estimated. Figure 6 demonstrates that the GPS gives as accurate prediction as the sample estimates (which are based on 100 samples) using only 10 training locations on the support of θ_1.

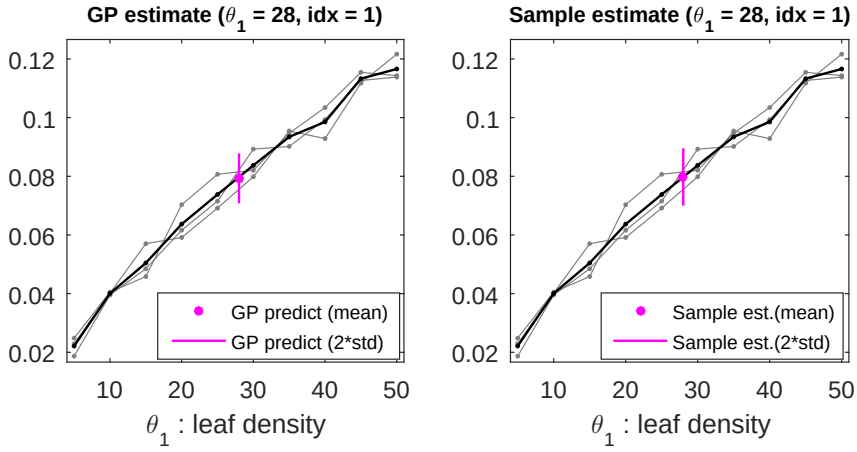

Fig. 6 A one-dimensional demonstration of the GP prediction using the sonar-foliage simulator. Here, we have fixed $\theta_2 = 0.017$ and $\theta_3 = 45$, and treated θ_1 as the unknown parameter. Left panel: the gray lines are the first wavelet coefficient of $m = 3$ simulated echo envelopes at $A = 10$ grid points on the domain $[5, 50]$; the black lines are the average of the three gray lines; the magenta dot and line are the predictive mean and the confidence interval ($mean \pm 2\,std$) calculated using the GPS. Right panel: the gray lines and the black lines are the same as the left panel. The magenta dot is the sample estimate of the mean, and the magenta bar is the confidence interval based on 100 echoes sampled directly from the simulator.

3. Based on the GPS, the likelihoods $\pi_\varepsilon(\bar{d}_y^j \mid \theta^*)$ and $\pi_\varepsilon(\bar{d}_y^j \mid \theta)$ can be approximated by $N(\bar{d}_y^j | \mu_{\theta^*}^{j,\dagger}, \sigma_j^2 + \varepsilon^2)$ and $N(\bar{d}_y^j | \mu_\theta^{j,\dagger}, \sigma_j^2 + \varepsilon^2)$ respectively, where $(\mu_{\theta^*}^{j,\dagger}, \mu_\theta^{j,\dagger})$ is a sample from $N(\mu^j_{(\theta^*,\theta)|\Theta}, \Sigma^j_{(\theta^*,\theta)|\Theta})$. The acceptance probability of the MCMC can be calculated by

$$\alpha(\theta^* | \theta) = \min\left\{1, \frac{\pi(\theta^*)\prod_{j=1}^J N(\bar{d}_y^j | \mu_{\theta^*}^{j,\dagger}, \sigma_j^2 + \varepsilon^2) q(\theta | \theta^*)}{\pi(\theta)\prod_{j=1}^J N(\bar{d}_y^j | \mu_\theta^{j,\dagger}, \sigma_j^2 + \varepsilon^2)\, q(\theta^* | \theta)}\right\}. \tag{7}$$

Note that if the function $f_j(\cdot)$ is known, we can replace $\mu_{\theta^*}^{j,\dagger}$ and $\mu_\theta^{j,\dagger}$ by the true values of $f_j(\theta^*)$ and $f_j(\theta)$ respectively, in which case $\alpha(\theta^* | \theta)$ is a deterministic value. However, since we have used GPS, the randomness of $\mu_{\theta^*}^{j,\dagger}$ and $\mu_\theta^{j,\dagger}$ introduces uncertainty to $\alpha(\theta^* | \theta)$. This uncertainty may cause an error for decision-making in

the MCMC algorithm. Therefore, we need to control the uncertainty so that the probability of making a wrong decision based on $\alpha(\theta^*|\theta)$ is reasonably low. We discuss this issue in Section 3.4.

3.4 Control the Uncertainty of Decision-making in wABC using GPS

The control of uncertainty in the GPS-based MCMC algorithm serves two purposes: to control the error rate of making decisions (e.g., the decision of accepting/rejecting the proposed θ^*) based on GPS in the MCMC algorithm, and to provide a strategy of refining the GPS of the simulator. The main idea is to keep adding training data to the GPS at each iteration until the expected probability of making the wrong decision is less than a pre-specified threshold ξ (e.g., $\xi = 0.3$).

In particular, since $\mu_{\theta^*}^{j,\dagger}$ and $\mu_{\theta}^{j,\dagger}$ are random samples from the GPS, α in equation (7) is a random variable. In stead of making decisions based on one α value, we produce L samples $\{\alpha^{(l)}, l = 1,\ldots,L\}$, and calculate a summary statistic ζ from it. We will accept θ^* if $u < \zeta$ and reject θ^* if $u \geq \zeta$. Here, $u \sim \text{Unif}(0,1)$.

Now we can calculate the probability of making a mistake following the above decision rule. If $u < \zeta$, we will accept θ^*, and this will be a wrong decision if indeed $\{u > \alpha\}$, in which case we should reject θ^*. The probability that this situation appears is $1_{\{u<\zeta\}} Pr(\{u > \alpha\})$. Similarly if $u \geq \zeta$, the probability of making a wrong decision is $1_{\{u\geq\zeta\}} Pr(\{u \leq \alpha\})$. Therefore, given a value of u, the overall probability of making an error is

$$W_u(\alpha) = 1_{\{u<\zeta\}} Pr(\{u > \alpha\}) + 1_{\{u\geq\zeta\}} Pr(\{u \leq \alpha\}), \quad u \sim \text{Unif}(0,1).$$

We can further integrate out u from the above conditional error function to obtain the marginal probability of making an error, i.e.,

$$W(\alpha) = \int_0^1 W_u(\alpha) du,$$

The above error probability $W(\alpha)$ is minimized when $\zeta = median(\alpha)$; a detailed argument can be found in the Section 3.1 of [1] and the reference therein.

The above result enables us to control the probability of making a wrong decision in the Step 3 in Section 3.3 by calculating α L times, each with different samples of $\mu_{\theta^*}^{j,\dagger}$ and $\mu_{\theta}^{j,\dagger}$. This calculation is very efficient since obtaining L samples from a multivariate normal distribution is fast. Based on the L samples of α, we set ζ to be the sample median of α and calculate $W(\alpha)$ numerically. If $W(\alpha) > \xi$, we will add more training points (i.e., creating a denser grid on the support of θ) and repeat Step 2-3 in Section 3.3 again, until $W(\alpha) \leq \xi$. We finally accept the proposed θ^* if $u < \zeta$ for a random u sampled from $\text{Unif}(0,1)$. This completes one iteration of the MCMC. The above adaptive strategy allows us to adjust for the training points

for the GPS so that the probability of making a wrong decision is controlled during each MCMC iteration.

4 The Algorithm and Parameter Settings

We describe the algorithm for the proposed wABC approach in the context of foliage echo data. Detailed steps are described in Algorithm 1. Algorithm 1 is an approximate MCMC algorithm because we have used GPS to approximate the simulator. These samples are used to approximate samples from the simulator. With GPS, we only need to call the simulator Δ times at each iteration under the condition $W(\alpha) > \xi$. As more training points are added, the GPS will become more reliable. Eventually, there will be no need to call the simulator at all during the MCMC iterations.

Algorithm 1: An MCMC Algorithm for wavelet-based ABC using GPS.

Input: $\mathbf{Y}$, A, Δ, θ, ε, ξ, $q(\cdot \mid \cdot)$, $\{\sigma_j^2\}$, $\pi(\theta)$, m, N, $k(\cdot,\cdot)$, the simulator.
Step 1: Perform wavelet decomposition and compression on $\mathbf{Y}$ to get $\bar{\mathbf{D}}_y$.
Step 2: Create a grid Θ of size A. Generate initial training points $\mathbf{X}$ from the simulator at each grid point in Θ. Perform wavelet decomposition and compression on each $\mathbf{X}$ to get $\{(\theta_i, \bar{d}_{x,i}^j), i = 1,\ldots,A\}$ for $j = 1,\ldots,J$.
Step 3: Run the following MCMC iterations.
for $i = 1$ *to* N **do**
  Propose θ^* from $q(\theta^* \mid \theta)$;
  while $W(\alpha) > \xi$ **do**
    Step 3.1: Calculate the mean and covariance for (θ^*, θ) following (5)-(6) for all j, $j = 1,\ldots,J$;
    Step 3.2: Generate L samples of $\mu_{\theta^*}^{j,\dagger}$ and $\mu_{\theta}^{j,\dagger}$ and calculate $\{\alpha^{(l)}, l = 1,\ldots,L\}$ using equation (7);
    Step 3.3: Set $\zeta = \text{median}(\{\alpha^{(l)}, l = 1,\ldots,L\})$ and calculate the probability of making a wrong decision $W(\alpha)$;
    if $W(\alpha) > \xi$ **then**
      Add Δ grid points to Θ, generate new training data at each newly added grid point. Add these points to the existing training points;
    end
  end
  Sample $u \sim \text{Unif}(0,1)$;
  if $u < \zeta$ **then**
    Set $\theta = \theta^*$;
  end
  Save θ;
end
Output: N posterior samples of θ.

For the numerical stability of the algorithm and the convenience of setting parameters, we recommend to rescale the compressed data $\widetilde{\mathbf{D}}_y$ and the simulated features

$\widetilde{\mathbf{D}}_x$ using a common set of constants so that all values are in a similar scale (e.g., $[-1,1]$). The scaling constants can be estimated from the observed data (e.g., using the minimum and maximum of $\mathbf{d}_y^j$ for each j). Similarly, in the GPS calculation, we recommend to scale all θ parameters to a common range (e.g., $[0,1]$).

Parameter settings. There are two types of model parameters we need to specify in the wABC algorithm—those in the GPS and those in the MCMC algorithm. Generally speaking, we suggest to determine parameters in GPS by checking the GP prediction at some θ values, so that the resulting GP prediction is comparable with that obtained by directly sampling from the simulator. Figure 6 provides an example of such comparison. Furthermore, we suggest to tune parameters in the MCMC algorithm by controlling the expected performance of the algorithm, such as the acceptance rate of the Metropolis-hastings sampler. In what follows, we introduce some specific guidelines.

The parameter ε in equation (7) is a small value that controls the expected discrepancy between simulated and observed data. We suggest to set a small value (e.g., 1e-4) for ε. In the GPS MCMC algorithm, it is possible to set $\varepsilon = 0$ as done by [1]. The parameters $\{\sigma_j^2\}$ in (4) control the noise level in the GP regression. We found that these parameters may substantially influence the predictive covariance of the GPS, i.e., the covariance in (6). A reasonable way to determine $\{\sigma_j^2\}$ is to take the empirical variance of $\mathbf{d}_x^j$ (calculated across the m replicates of $\widetilde{\mathbf{D}}_x$) and average them across all grid points in Θ. The parameters in the GP kernel $k(\cdot,\cdot)$ also play important roles in determining the predictive mean and covariance of the GPS. We have used the squared exponential kernel $k_j(\theta,\theta^*) = \phi_j^2 \exp\{-||\theta-\theta^*||^2/(2\tau_j^2)\}$ for each j. In the foliage-echo data analysis, we have scaled the θ parameters to $[0,1]$ and scaled all $\bar{d}_x^j, j = 1,\ldots,J\}$ to $[-1,1]$. Under these setups, we found that setting $\phi_j \equiv 0.1$ and $\tau_j \equiv 0.4$ is a reasonable choice. In practice, we recommend the users to start with the one-parameter settings (i.e., fixing all other parameters) and plot the predictive error bar like shown in Figure 6. This helps visualize the effect of the parameter setups. The parameters ξ, m, N, A, and Δ can be tuned based on the computation speed and the acceptance rate of the MCMC algorithm.

In general, the accuracy of the posterior estimation can be improved by increasing the sample size in data $\mathbf{Y}$, reducing the threshold ξ for the probability of making an error in the MH sampler, increasing the size of the training grid for GPS, and increasing the number of training samples m at each GP training grid.

5 The Analysis of Simulated Foliage-echo Data

While our ultimate goal is to apply wABC on real foliage-echo data collected under experimental or natural environments, at this stage, the real data has not yet been made available. Therefore, in this analysis, we will only provide the parameter estimation result based on echoes simulated from the foliage-echo simulation model described in the Appendix. Because the true parameters are known in this simula-

tion setup, our analysis provides the proof-of-concept for the feasibility of wABC for complex systems.

We applied the proposed wABC approach to a set of foliage-echo data simulated from the sonar-foliage simulator. The data consists of $n = 100$ echo envelope signals sampled independently from the simulator under the true parameter $(\theta_1, \theta_2, \theta_3) = (30, 0.017, 45)$. We aim to solve the inverse-problem by estimating the three underlying parameters based on the 100 echo envelopes while assuming that the domains of the parameters are $\theta_1 \in [5, 50]$, $\theta_2 \in [.005, .05]$, and $\theta_3 \in [1\text{e-}4, 90]$.

We applied the wavelet transformation to each echo envelope using Daubechies wavelets with the maximal number of vanishing moments being 12 (i.e., db12). The number of resolution levels is set to be $J = 20$, and the boundary extension mode is set to be periodic. The wavelet decomposition transforms each echo envelope from the time domain (with $24,000$ measurement points) to the wavelet domain (with $24,008$ wavelet coefficients). We further applied wavelet compression by retaining $\delta_1 = 0.999$ of the total energy. This reduces the dimension of the wavelet coefficients from $24,008$ to 992. We then applied MCMC with GPS using Algorithm 1. We adopted a random walk proposal by setting the proposal distribution $q(\theta^* \mid \theta)$ to be a truncated log-normal with a scale parameter 0.05. To train the GPS, we segmented the domain of $(\theta_1, \theta_2, \theta_3)$ using a $10 \times 10 \times 10$ equally-spaced grid. This gave a total of 1000 training points for the GPS. The number of repeated samples in $\mathbf{X}$ on each grid point was set to be $m = 3$. The kernel parameters for the Gaussian process kernel function were set to be $\phi_j \equiv 0.1$, $\tau_j \equiv 0.4$. The ε parameter in the MCMC-ABC was set to be 1e-4 and the ξ parameter in the GPS procedure was set to be 0.3. These setups resulted in an acceptance rate of 35% in the MCMC MH sampler. We monitored the behavior of the posterior samples by checking the trace plots and the autocorrelation plots. We tested the convergence of the chains by calculating the Geweke's Z-statistics [8]. We ran $30,000$ MCMC iterations and took the first $10,000$ iterations as the burn-in period. Summary statistics of the parameter estimation, including the posterior means and the 95% credible intervals (CIs), are listed in Table 1. Table 1 shows that all three CIs cover the true values of the parameters.

Table 1 The posterior estimation for the three parameters in the foliage-echo data.

	θ_1	θ_2	θ_3
Meaning	density in 3-d	mean radius	mean orientation
Unit	counts per m^3	meter	degree
Domain	[5, 50]	[.005, .05]	[1e-4, 90]
True value	30	.017	45
Post. mean	28.45	.018	42.57
Post. CIs	[17.1, 41.9]	[.017, .026]	[10.1, 72.7]

We further summarized the posterior distribution of parameters using 1-d and 2-d marginal kernel density estimations. In Figure 7, we plot the heatmaps of the 2-d kernel density estimations for each pair of the parameters. The gray dots on the

heatmaps are the scatter plots of the posterior samples (a total of $15,000$ samples after the burnin period). The white cross sign on the heatmaps mark the true values of the parameters. The histograms on the top and right-hand side of each heatmap show the marginal distributions of the parameters (superimposed by the 1-d density estimations). The red vertical bar in each histogram indicates the location of the posterior mean, and the red dashed bars indicate the 95% credible interval.

From Figure 7, we observe that the posterior distributions of the parameters demonstrate skewed, multi-modality shapes. In particular, the marginal distribution of the leaf size is skewed to the right, and the leaf orientation demonstrates two modes, one near 10 degree and the other near 40 degree. Furthermore, the 95% CIs for the leaf density and the orientation are fairly wide. Wide CIs indicate high uncertainty in the point estimates. These results are not a surprise, because they reflect several characteristics of the foliage-echo simulation. First, the echo signals are highly stochastic—using 100 echoes samples to recover the statistical properties of the foliages is a challenging task. Second, the multi-modal behavior of the posterior distribution reflects the non-identifiability nature of the inverse-problem, i.e., different combinations of the leaf density, size, and orientation could result in similar reflection behavior of the sound wave. Therefore, the solution to the inverse-problem is not unique. Despite these challenges, our proposed wABC still provide a comprehensive view for the distributions of the underlying parameters under a moderate number of samples—a result that is intractable if using any other existing statistical approaches. These results demonstrate the promise of solving ill-posed inverse-problems even when the data is highly stochastic and of high-dimension.

6 Discussion

We have proposed a general simulation-based approach called wABC to estimate the parameters of a complex system with functional data outputs. The proposed method relies on simulating from the complex system to estimate the parameters of interest, which avoids the difficulty of specifying the intractable likelihood. We accommodate functional data measured on a dense, high-dimensional grid by combining wavelet decomposition with compression, and achieve scalable computation using a Gaussian process surrogate to the simulator. Our inference is based on posterior samples of the underlying parameters which can be used to recover the joint distributions of all parameters.

The proposed wABC approach is generally applicable to a large family of inverse-problems associated with complex systems, such as solving differential equations based on noisy data and estimating parameters of a biological system. However, it requires a "simulator" to generate pseudo-data. The simulator needs to resemble the real system with sufficient accuracy. Otherwise, even if the wABC is tuned to perform well with simulated data, it may fail on real-word data.

While the GPS has the benefit of avoiding repeatedly calling the simulator, the computation of GP may become inefficient when the number of grid points goes be-

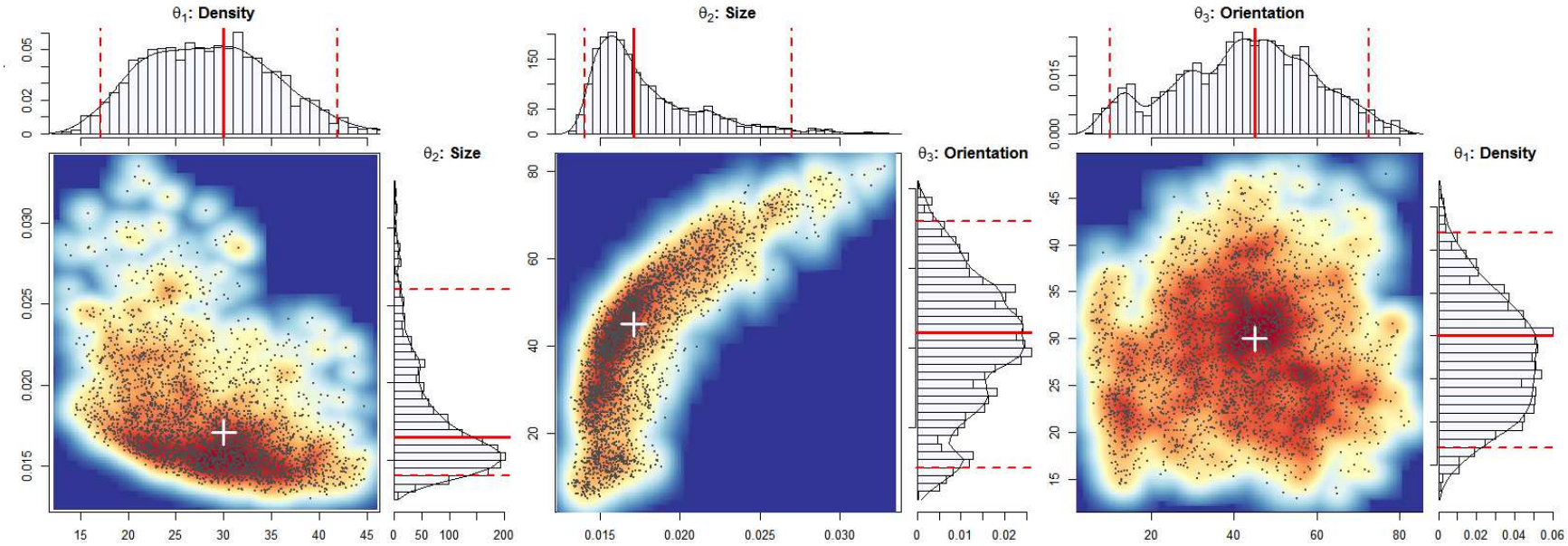

Fig. 7 Results for foliage-echo data analysis. The heatmaps of the 2-d kernel density estimations for each pair of parameters. Left: θ_1 versus θ_2; middle: θ_2 versus θ_3; right: θ_3 versus θ_1. The gray dots on the heatmaps are the scatter plots of the posterior samples. The white cross symbol on the heatmaps marks the true parameter values. The histograms on the top and the right-hand side of each heatmap are the marginal distributions (histograms superimposed by the 1-d kernel density estimations) for each parameter. The red vertical bar in each histogram indicates the location of the posterior mean, and the red dashed bars indicate the 95% credible interval.

yond 1000. The main difficulty comes from the evaluation of the inverse covariance matrix in the predictive calculation, i.e., equations (5)-(6). Our future work involves replacing the GPS by a local-GPS, i.e., using only a portion of the training points to predict the mean and standard deviation at (θ^*, θ). A promising strategy is to use the K-nearest neighbor approach. Similar ideas have been adopted by [9] and [10] in GP regression.

While we have focused on systems with highly-stochastic functional outputs, the proposed framework is also suitable for deterministic systems in which the simulator yields a deterministic functional output subject to random measurement error. An example is the LIDAR data introduced in Section 1. In these situations, we just need to set $n = 1$ and $m = 1$ in wABC. These problems are often easier to solve than the stochastic systems considered here.

Although ABC brings substantial convenience by enabling likelihood-free inference, it is well-recognized that ABC suffers from "curse of dimensionality" when the number of parameters increases. That is, it becomes more difficult to accept a proposed parameter θ^* as the dimension of the parameter space increases [26]. We adopted an MCMC algorithm in this paper, which has been proposed to attenuate the low acceptance rate issue. However, it remains generally true that a larger number of parameters is more expensive to be estimated using ABC-based approaches.

Though a real data analysis is not included in this paper, our simulation study provides a solid validation of the statistical component of the method. Given that the proposed method performs well under simulated setting, the only situation under which it will fail in a real data analysis is when the physical model (i.e., the simulator) does not describe the scenario of the real data properly. If that happens, one either adjusts the way to collect real data or modifies the physical model.

Finally, we note that it remains a future work to develop the theoretical properties of the proposed wABC approach. In particular, it is of interest to demonstrate the convergence of the MCMC to a stationary distribution, and show that the resulting stationary distribution approximates the true posterior distribution with a bounded error. These theoretical investigations may be done following the arguments/hints in [12] and [1].

Appendix: More Details of the Foliage-echo Simulator

In this appendix, we provide more details about the simulation model. Let $Y(t)$ denote a random echo signal to be simulated based on the sonar scene described in Section 2. We will simulate $Y(t)$ discretely, i.e., simulate the vector $\mathbf{y} = (y_1, \ldots, y_v)^T$, a discretized version of $Y(t)$. Here, the sampling frequency of $\mathbf{y}$ is 400kHz. To achieve this, we first simulate $\widehat{\mathbf{y}} = (\widehat{y}_1, \ldots, \widehat{y}_{v'})^T$, which is the Fourier transform of $\mathbf{y}$ in the frequency domain. We then apply the inverse fast Fourier transform to $\widehat{\mathbf{y}}$ to obtain $\mathbf{y}$.

Each component in $\widehat{\mathbf{y}}$ corresponds to a fixed frequency. We denote by f_k the frequency corresponding to $\widehat{y}_k$ (the kth component of $\widehat{\mathbf{y}}$). In this simulation, we mimic the frequency range of a horseshoe bat's echolocation call [28] and only simulate

the Fourier components corresponding to frequencies in the range of $[60,80]$ kHz. All other Fourier components are set to zero. For $f_k \in [60,80]$ kHz, $\widehat{y}_k$ is a complex number in the form of $\widehat{y}_k = \sum_{i=1}^{s} A_{k,i}\cos(\phi_{k,i}) + j\sum_{i=1}^{s} A_{k,i}\sin(\phi_{k,i})$, where j denotes the imaginary unit, s denotes the number of leaves considered, $A_{k,i}$ is the amplitude at frequency f_k for the ith leaf, and $\phi_{k,i}$ is a phase delay parameter at f_k for the ith leaf.

The key of our simulation is the calculation of $\{(A_{k,i},\phi_{k,i}), k = 1,\ldots,\nu', i = 1,\ldots,s\}$ based on the physical laws of sound transmission and reflection. This calculation is performed through four steps:

1. **Simulate the foliage scene.** From the input parameters $(\theta_1,\theta_2,\theta_3)$, simulate the total number of leaves (denoted by s_0), the radii of leaves $\{a_i\}$, the 3-d coordinates of the leaf centers $\{(x_i,y_i,z_i), i = 1,\ldots,s_0\}$, and the incident angles $\{\beta_i, i = 1,\ldots,s_0\}$ following the description in Section 2.
2. **Select leaves that contribute to echo.** Based on the locations of the leaves and the sonar, we calculate the sonar's beampattern gains (i.e., the spatial distribution of sound pressure) at all leaves, and filter out those leaves that have small gain values. Therefore, only leaves at locations with large enough sonar gain values are used to simulate **y**. We denote the number of leaves passing this filter by s.
3. **Calculate amplitudes.** The parameter $A_{k,i}$ represents the amplitude corresponding to the wave reflected from the ith leaf at frequency f_k. It is calculated based on the formula:

$$A_{k,i} = S(az_i, el_i, f_k, r_i)L_i(\beta_i, a_i, f_k)\frac{\lambda_k}{2\pi r_i^2}. \tag{8}$$

   Below, we will explain the meaning of each factor in (8):

   a. The factor $S(az_i, el_i, f_k, r_i)$ denotes the sonar beampattern, a function that describes the spatial distribution of the power density of the emitted wave. The arguments (az_i, el_i) denote the azimuth and elevation angles of the line that connects the origin (i.e., the sonar) and the ith leaf center, and the argument r_i denotes the distance between the sonar and the ith leaf center. Here, (az_i, el_i) and r_i can be directly calculated from the leaf center coordinates (x_i,y_i,z_i). For a given sonar, $S(\cdot)$ is assumed to be known. In this study, we used a Gaussian function to approximate the sonar beampattern. The parameters of the Gaussian function are determined using empirical data. In particular, the Gaussian function parameters are determined by three variables: the beamwidth of the sonar beampattern (-3 decibel), the direction that sonar faces, and the peak amplitude of the sonar beampattern.
   b. The factor $L_i(\beta_i, a_i, f_k)$ denotes the beampattern of the ith leaf, which describes the spatial distribution of the power density of the reflected wave at the ith leaf. Here, β_i is the incident angle of the ith leaf, a_i is the radius of the ith leaf, and f_k is the kth frequency. The leaf beampattern can be calculated using complicated physical equations [3]. In this study, we approximate the leaf beampattern using

$$L_i(\beta_i, a_i, f_k) = P_1(c(f_k, a_i))\cos(P_2(c(f_k, a_i))\beta_i),$$

where $c(f_k, a_i) = 2\pi a_i f_k / v$, $v = 340$ (meters per second) denotes the speed of sound, $P_1(c(f_k, a_i)) = 0.5003c^2 + 0.6867$, and $P_2(c(f_k, a_i)) = 0.3999c^{-0.9065} + 0.9979$. The functions $P_1(\cdot)$ and $P_2(\cdot)$ are nonlinear regression functions estimated based on data obtained from numerical evaluation [2].

c. In the factor $\lambda_k/(2\pi r_i^2)$, r_i is the distance between the sonar and the ith leaf center, and λ_k is the wavelength of the emitted sound wave corresponding to the frequency f_k. Here, λ_k is a known constant.

4. **Calculate phase delays.** The phase delay parameters $\{\phi_{k,i}\}$ reflect the phase change at leaf i and frequency f_k due to wave propagation. After waves travel r_i meters, the phase delay becomes $2\pi r_i/\lambda_k$. As it is a round trip for the sound to travel from sonar to leaf and from leaf to sonar, the phase delay due to propagation is $4\pi r_i/\lambda_k$. Another part that contributes to the phase delay is the phase shift after the wave strikes the leaf. The phase shift depends on the frequency f_k, the leaf radius a_i, and the incident angle β_i. To make the computation efficient, we estimate the phase shift by fitting a nonlinear regression based on data obtained from numerical evaluation [2], which gives

$$\text{Phase\_shift}(f_k, a_i, \beta_i) = \text{erf}(\text{PA}(c(f_k, a_i))(1.57 - \beta_i)) - 2.6343,$$

where $\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$, $\text{PA}(c(f_k, a_i)) = 0.9824c(f_k, a_i)^{0.3523} - 0.9459$, and $c(f_k, a_i) = 2\pi a_i f_k / v$. Based on these results, the phase delay can be calculated by

$$\phi_{k,i} = -\frac{4\pi r_i}{\lambda_k} - \text{Phase\_shift}(f_k, a_i, \beta_i) \tag{9}$$

Steps 1–4 provide the values of $\{(A_{k,i}, \phi_{k,i})\}$, based on which we can calculate the frequency domain vector $\widehat{\mathbf{y}}$. The final time domain signal $\mathbf{y}$ is calculated by using the inverse fast Fourier transform. Before applying the transform, we also applied a Hann window function to weight $\widehat{\mathbf{y}}$, which helps minimize the signal side lobes (unwanted ripples) in the resulting time domain signal.

References

[1] GPS-ABC: gaussian process surrogate approximate bayesian computation. Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI) (2014)

[2] Adelman, R., Gumerov, N.A., Duraiswami, R.: Software for computing the spheroidal wave functions using arbitrary precision arithmetic. CoRR (2014). URL http://arxiv.org/abs/1408.0074

[3] Bowman, J., Senior, T., Uslenghi, P.: Electromagnetic and acoustic scattering by simple shapes. Hemisphere Publishing Corporation, New-York (1987)

[4] Cardot, H.: Nonparametric regression for functional responses with application to conditional functional principle component analysis. Available online: http://www.lsp.ups-tlse.fr/Recherche/Publications/2005/car01.pdf (2005)
[5] Chiou, J., Müller, H., Wang, J.: Functional quasi-likelihood regression models with smooth random effects. J. Royal Statist. Soc. Ser. B **65**, 405–423 (2003)
[6] Didelot, X., Everitt, R.G., Johansen, A.M., Lawson, D.J.: Likelihood-free estimation of model evidence. Bayesian Anal. **6**(1), 49–76 (2011). DOI 10.1214/11-BA602. URL http://dx.doi.org/10.1214/11-BA602
[7] Ferraty, F., Vieu, P.: Nonparametric Functional Data Analysis. Springer-Verlag (2006)
[8] Geweke, J.: Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. Bayesian Statistics 4 (1992)
[9] Gramacy, R.B., Apley, D.W.: Local gaussian process approximation for large computer experiments. J. Comput. Graph. Stat. **24**(2), 561–578 (2015)
[10] Gramacy, R.B., Haaland, B.: Speeding up neighborhood search in local Gaussian process prediction. ArXiv e-prints (2014)
[11] Horváth, L., Kokoszka, P.: Inference for Functional Data with Applications. Springer Verlag (2012)
[12] Korattikara, A.B., Chen, Y., Welling, M.: Austerity in MCMC land: cutting the Metropolis-Hastings budget. In: Proceedings of the 31st International Conference on Machine Learning, Cycle 1, *JMLR Proceedings*, vol. 32, pp. 181–189. JMLR.org (2014)
[13] Lehmann, E.L., Casella, G.: Theory of Point Estimation, 2nd edn. Springer (1998)
[14] Lu, T., Liang, H., Li, H., Wu, H.: High dimensional odes coupled with mixed-effects modeling techniques for dynamic gene regulatory network identification. J. Am. Statist. Ass. **106**(496), 1242–1258 (2011)
[15] Marin, J.M., Pudlo, P., Robert, C.P.R., Ryder, R.J.: Approximate bayesian computational methods. Stat. Comput. **22**, 1167–1180 (2012)
[16] Morris, J.S.: Functional regression. Annu. Rev. Stat. Appl. **2**, 321–359 (2015)
[17] Pritchard, J.K., T., S.M., A., P.L., W., F.M.: Population growth of human y chromosomes: a study of y chromosome microsatellites. Mol. Biol. Evol. **16**, 1791–1798 (1999)
[18] Ramsay, J.O., Hooker, G., Campbell, D., Cao, J.: Parameter estimation for differential equations: a generalized smoothing approach. J. Royal Statist. Soc. Ser. B **69**(5), 741–796 (2007)
[19] Ramsay, J.O., Li, X.: Curve registration. J. Royal Statist. Soc. Ser. B **60**(2), 351–363 (1998)
[20] Ramsay, J.O., Silverman, B.W.: Functional Data Analysis. Springer-Verlag, New York (1997)
[21] Rice, J.A., Silverman, B.W.: Estimating the mean and covariance structure nonparametrically when the data are curves. J. Royal Statist. Soc. Ser. B **53**, 233–243 (1991)

[22] Sadegh, M., Vrugt, J.A.: Approximate bayesian computation using markov chain monte carlo simulation: Dream(abc). Water Resources Research **50**(8), 6767–6787 (2014). DOI 10.1002/2014WR015386
[23] Sanders, J., Kandrot, E.: CUDA by Example: An Introduction to General-Purpose GPU Programming, 1st edn. Addison-Wesley Professional (2010)
[24] Scheipl, F., Staicu, A.M., Greven, S.: Functional additive mixed models. J. Comput. Graph. Stat. **24**, 477–501 (2014)
[25] Tang, R., Müller, H.G.: Pairwise curve synchronization for functional data. Biometrika **95**(4), 875 (2008)
[26] Turner, B.M., Van Zandt, T.: A tutorial on approximate bayesian computation. Journal of Mathematical Psychology **56**, 69–85 (2012)
[27] Vanderelst, D., Steckel, J., Boen, A., Peremans, H., Holderied, M.W.: Place recognition using batlike sonar. eLife **5**, e14,188 (2016). DOI 10.7554/eLife.14188
[28] Vaughan, N., Jones, G., Harris, S.: Identification of british bat species by multivariate analysis of echolocation call parameters. Bioacoustics **7**(3), 189–207 (1997)
[29] Wang, J.L., Chiou, J.M., Müller, H.: Review of functional data analysis. Annu. Rev. Statist. pp. 1–41 (2015)
[30] Wegmann, D., Leuenberger, C., Excoffier, L.: Efficient approximate bayesian computation coupled with markov chain monte carlo without likelihood **182**(4), 1207–1218 (2009). DOI 10.1534/genetics.109.102509
[31] Wikipedia: Sonar — Wikipedia, the free encyclopedia (2017). URL https://en.wikipedia.org/wiki/Sonar. [Online; accessed 20-May-2017]
[32] Xun, X., Cao, J., Mallick, B., Maity, A., Carroll, R.J.: Parameter estimation of partial differential equation models. J. Am. Statist. Ass. **108**(503), 1009–1020 (2013)
[33] Yang, J., Zhu, H., Choi, T., Cox, D.D.: Smoothing and meanccovariance estimation of functional data with a bayesian hierarchical model. Bayesian Anal. **11**(3), 649–670 (2016)
[34] Yao, F., Müller, H.G., Wang, J.L.: Functional data analysis for sparse longitudinal data. J. Am. Stat. Assoc. **100**, 577–590 (2005)
[35] Zhang, X., Cao, J., Carroll, R.J.: Estimating varying coefficients for partial differential equation models. Biometrics (2017)
[36] Zhu, H., Brown, P.J., Morris, J.S.: Robust, adaptive functional regression in functional mixed model framework. J. Am. Statist. Ass. **495**, 1167–1179 (2011)