

A Weighted Surrogate Model for Spatio-Temporal Dynamics with Multiple Time Spans Applications for the Pollutant Concentration of the Bai River

Huan, Yue; Tian, Yubin; Wang, Dianpeng

DOI

[10.3390/math10193585](https://doi.org/10.3390/math10193585)

Publication date

2022

Document Version

Final published version

Published in

Mathematics

Citation (APA)

Huan, Y., Tian, Y., & Wang, D. (2022). A Weighted Surrogate Model for Spatio-Temporal Dynamics with Multiple Time Spans: Applications for the Pollutant Concentration of the Bai River. *Mathematics*, 10(19), Article 3585. <https://doi.org/10.3390/math10193585>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Article

A Weighted Surrogate Model for Spatio-Temporal Dynamics with Multiple Time Spans: Applications for the Pollutant Concentration of the Bai River

Yue Huan ^{1,2}, Yubin Tian ¹ and Dianpeng Wang ^{1,*}¹ School of Mathematics and Statistics, Beijing Institute of Technology, Beijing 100081, China² Delft Institute of Applied Mathematics, Delft University of Technology, 2628 CD Delft, The Netherlands

* Correspondence: wdp@bit.edu.cn

Abstract: Simulations are often used to investigate the flow structures and system dynamics of complex natural phenomena and systems, which are significantly harder to obtain from experiments or theoretical analyses. Surrogate models are employed to mimic the results of simulations by reducing computational costs. In order to reduce the amount of computational time consumed, a novel framework for building efficient surrogate models is proposed in this work. The novelty lies in that the new framework runs simulations using the different simulation time spans for different inputs and builds a comprehensive surrogate model through the fusion of non-homogeneous spatio-temporal data by integrating the temporal and spatial correlations in parametric space. This differs from the existing works in the literature, which only consider the situation of spatio-temporal data with a consistent time span during simulations under different inputs. Some simulation studies and real data analysis concerning the pollution of the river in the Sichuan Province of China are used to demonstrate the superior performance of the proposed methods.



Citation: Huan, Y.; Tian, Y.; Wang, D. A Weighted Surrogate Model for Spatio-Temporal Dynamics with Multiple Time Spans: Applications for the Pollutant Concentration of the Bai River. *Mathematics* **2022**, *10*, 3585. <https://doi.org/10.3390/math10193585>

Academic Editors: Nancy Joy Lim, Natalia Nikolova and Mark Kelbert

Received: 19 August 2022

Accepted: 28 September 2022

Published: 1 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: spatio-temporal data; proper orthogonal decomposition; cokriging; prediction

MSC: 62K99; 62P12

1. Introduction

In practice, partial differential equations (PDEs) are often employed to study complex natural phenomena and engineering systems in many fields, such as meteorology, turbulence, and aircraft design. Numerical methods, such as the finite element algorithm, are used to solve PDEs, referred to in the literature as simulators. High-fidelity simulations have been used for investigating system dynamics and flow structures. However, it is time-consuming to run the corresponding simulations. Modeling and estimating the spatio-temporal dynamics over a wide parametric space are important challenges that are yet to be overcome. Surrogate models are widely utilized to mimic the results of simulators with significantly reduced computational costs.

There have been many previous works building surrogate models for spatio-temporal dynamics in the literature. Xiao et al. [1] present a non-intrusive subdomain POD-TPWL (SD POD-TPWL) method through integral domain decomposition (DD), proper orthogonal decomposition (POD, which was first proposed by [2]), radial basis function (RBF) interpolation, and trajectory segmentation linearization (TPWL). Ioannidis et al. [3] proposed a graph-aware kernel kriged Kalman filtering (KKF) method accounting for the spatio-temporal variations. Nguyen et al. [4] and Shi et al. [5] modeled spatio-temporal dynamics using mixed-effects models. These methods only pay attention to spatio-temporal dynamics under fixed input parameters. For example, the authors of [1] investigated the spatio-temporal performance of the oil–water reservoir system at fixed values of initial

pressure, initial saturation, etc., instead of studying the influence of input parameters on the spatio-temporal dynamics.

Several works have proposed modeling the relationship between the spatio-temporal dynamics and the input parameters. Guo and Hesthaven [6] proposed a data-driven reduced basis (RB) method for parameterized spatio-temporal modeling problems, which reduced the dimensions of training data by POD and then built the regression-based error surrogate model. Yeh et al. [7] used POD to decompose the spatio-temporal system of the flow field in the vortex ejector and then trained the kriging model for the reduced data, which can predict the system dynamics under any input parameter. Chang et al. [8,9] proposed new surrogate models named kernel-smoothed proper orthogonal decomposition (KSPOD) and common kernel-smoothed proper orthogonal decomposition (CKSPOD) to emulate spatio-temporally evolving flows.

In order to build a more economical surrogate model, most of the above works used the POD, a widely used reduced order modeling (ROM) method, to represent the spatio-temporal data at a controlled loss of accuracy. From a data-driven point of view, Yeh et al. [7] sampled a set of input parameters $\Theta \triangleq \{\theta_1, \theta_2, \dots, \theta_c\}$ and ran simulations to generate the spatio-temporal data of the flow field in the vortex ejector $y = f(x, t; \theta)$ for each $\theta \in \Theta$. Then $f(x, t; \theta)$ is approximated into K common POD modes $\phi_k(x)$ with time-varying coefficients $\beta_k(t; \theta)$, that is

$$f(x, t; \theta) \approx \sum_{k=1}^K \beta_k(t; \theta) \cdot \phi_k(x), \tag{1}$$

with the reasonable expectation that the approximation becomes exact as $K = N$ [10], where $\phi_k(x)$ is a spatial basis function, and $\beta_k(t; \theta)$, $k = 1, 2, \dots, K$ is the k -th time-varying coefficient under input parameter θ and time t , which describes the relationship between $\phi_k(x)$ and $f(x, t; \theta)$ given θ and t . Given k and time step t_m , time-varying coefficients $\beta_k(t_m; \theta)$ can be seen as functions of the input variable θ . Hence, the kriging model is established for every $\beta_k(t_m; \theta)$, $m = 1, 2, \dots, M$ and $k = 1, 2, \dots, K$. For any new parameter θ^* in the parameter space Ω , the kriging model can give the prediction $\hat{\beta}_k(t_m; \theta^*)$. Then, the prediction of the system dynamics under the new parameter is given by

$$\hat{f}(x, t; \theta) = \sum_{k=1}^K \hat{\beta}_k(t; \theta) \cdot \phi_k(x). \tag{2}$$

Most of the existing works in the literature assume that the temporal resolution is fine [7] and build a surrogate model based on the same time span \mathcal{T} . Attention has not been paid to the time correlations; however, for some practical problems, there are correlations between different time steps, such as the incompressible fluid flow, see [6].

Let $[0, t_M]$ be the time span of interest to engineers. In practice, due to the limitations of computational time and resources, only a few parameters of the spatio-temporal data are simulated to the last time step t_M , and the others are only simulated to the M_i -th time steps t_{M_i} , $i = 1, 2, \dots, s$, $t_{M_1} < t_{M_2} < \dots < t_{M_s} = t_M$. Hence, the spatio-temporal data with different time spans need to be fused to build a comprehensive surrogate model. In this way, computational resources and time can be saved to simulate the spatio-temporal data under more parameters. Let $\mathcal{T}_i \triangleq \{t_1, t_2, \dots, t_{M_i}\}$, $i = 1, 2, \dots, s$, and $\mathcal{T} = \mathcal{T}_s$. Assume c is the number of parameters required to run the simulations for the traditional methods. Let $s = 2$. Let c_1 be the number of parameters under which the spatio-temporal data are simulated over \mathcal{T}_1 , and c_2 be the number of parameters under which the spatio-temporal data are simulated over \mathcal{T} . c_1 and c_2 can be chosen appropriately such that $c_1 \times t_{M_1} + c_2 \times t_M \approx c \times t_M$ and $c_1 + c_2 > c > c_2$. Thus, under the same computing resources, more input parameters, and their corresponding spatio-temporal data will be considered, which leads to a more accurate surrogate model. Take the case study of pollutant concentration in Section 4 as an example, for each input, the simulation will take

approximately 22 h on a computer with 72 Intel(R) Xeon(R) Gold 6254 CPU @ 3.10 GHz and 16 G memory. If the number of inputs is $c = 20$, it will take 440 h with the full time span for all of the inputs. If the number of inputs with full time span simulation is $c_1 = 10$ and the number of inputs with half time span simulation is $c_2 = 20$, it will also take approximately 440 h. However, we collect the data for approximately 30 different inputs, which could benefit the accuracy of the surrogate model.

Consider the pollution supervision of rivers as an example. A simulator is generally established for the hydrological situation of the entire river basin, the relevant indicators of the source of pollutants are used as input parameters, and certain spatio-temporal data are generated. In daily supervision, the spatio-temporal field under certain parameters in a certain period of time \mathcal{T}_1 is simulated through the simulator. However, when there is an emergency, a quick response of the flow structures over \mathcal{T}_2 , which exceeds \mathcal{T}_1 , is required. Therefore, it is necessary to quickly generate the spatio-temporal data over $\mathcal{T}_2 \setminus \mathcal{T}_1$ at fewer parameters and establish a surrogate model to determine the source of pollutants and predict the river's pollution distribution. For this case, a surrogate model is needed that uses different time spans for training data. In this work, we monitor pollution in the Bai River as an example to demonstrate the effectiveness of the proposed methods.

The objective of this paper is to propose a framework to build a surrogate model for fusing spatio-temporal data with multiple time spans. In order to reduce the amount of computational time consumed, the new framework is used to run simulations using different simulation time spans for different inputs, and to build a comprehensive surrogate model through the fusion of non-homogeneous spatio-temporal data by integrating temporal and spatial correlations.

The remainder of this paper is organized as follows. A new predictive surrogate model to fuse spatio-temporal data over different time spans is presented in Section 2. In Section 3, some simulations are carried out to illustrate the performance of the proposed methods when the training data have different time spans. In Section 4, real data analysis for the pollution of rivers in the Sichuan Province of China is given. Conclusions are drawn in Section 5.

2. Methodology

2.1. The Reverse Sequential Sampling Scheme

Usually, the spatio-temporal data are divided into N discrete spatial nodes $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ and M time steps $\mathcal{T} = \{t_1, t_2, \dots, t_M\}$. Suppose that there are s different time spans $\mathcal{T}_1 \subset \mathcal{T}_2 \subset \dots \subset \mathcal{T}_s = \mathcal{T}$, which will be used in the simulations. Let the end of time span \mathcal{T}_i be t_{M_i} , $i = 1, 2, \dots, s$ and assume that the time interval Δt is the same over the entire time span. For each time span \mathcal{T}_i , c_i input parameters are sampled from parameter space Ω , and the corresponding spatio-temporal data are generated through the simulator. For spatio-temporal data at $\mathcal{T}_s \setminus \mathcal{T}_{s-1} \triangleq \{t_{M_{s-1}+1}, t_{M_{s-1}+2}, \dots, t_{M_s}\}$, there are only c_s parameters available. For $\mathcal{T}_{s-1} \setminus \mathcal{T}_{s-2} \triangleq \{t_{M_{s-2}+1}, t_{M_{s-2}+2}, \dots, t_{M_{s-1}}\}$, the number of parameters augment $c_s + c_{s-1}$. Thus, there are $\sum_{i=1}^s c_i$ training parameters, which is at a maximum over the first time span $\mathcal{T}_1 = \{t_1, t_2, \dots, t_{M_1}\}$.

In order to ensure the prediction accuracy of the surrogate model, at each time point t_m , the selected parameters should have a good space-filling ability. Since the fewest training input parameters correspond to the spatio-temporal data at the time step $t \in \{t_{M_{s-1}+1}, t_{M_{s-1}+2}, \dots, t_{M_s}\}$, the selection of parameters over this time span should be prioritized. A feasible method is to uniformly sample c_s parameters from the parameter space Ω , which is denoted as Θ_s . Then, using sequential Latin hypercube designs (LHDs), such as the Quasi-LHD sequential sampling method [11,12] and maximum projection (MaxPro) LHD [13,14], c_{s-1} parameters are sampled from $\Omega \setminus \Theta_s$, the relative complement of Θ_s in Ω , to compose Θ_{s-1} , so that the parameters in $\Theta_s \cup \Theta_{s-1}$ have the space-filling property in Ω . Continue this method until the sampling of Θ_1 is finished. It is obvious that $\Theta_i \cap \Theta_j = \emptyset, 1 \leq i < j \leq \dots, s$. Let the j -th element in Θ_i be denoted as $\theta_{ij}, j = 1, 2, \dots, c_i$. Then, the parameters in $\Theta_i, i = 1, 2, \dots, s$, are used to simulate the spatio-temporal data

for $\mathcal{T}_i = \{t_1, t_2, \dots, t_{M_i}\}$. This method can sequentially generate spatio-temporal data with s different time spans for different input parameters, and the parameters have a good space-filling property in Ω and are uniform in each Θ_i . In this paper, we refer to this sampling procedure as the reverse sequential sampling scheme, in which the “reverse” means that the order in which the parameters are sampled is the reverse of the order in which the simulator is run. The data simulation approach in this paper adopts the reverse sequential sampling scheme.

As an example to illustrate this method, 1-dimensional parameter space and time space are considered. The data with three time spans $\mathcal{T}_1 = \{1, 2, \dots, 5\}$, $\mathcal{T}_2 = \{1, 2, \dots, 10\}$, $\mathcal{T}_3 = \{1, 2, \dots, 15\}$ and three parameter sets $\Theta_1, \Theta_2, \Theta_3 \in [0, 1]$ are generated. $\Theta_1 = [0.16, 0.25, 0.50, 0.84]$ was first sampled from $[0, 1]$, such that the elements in Θ_1 are uniform in $[0, 1]$. Next $\Theta_2 = [0.21, 0.34, 0.61, 0.80]$ was sampled from $[0, 1] \setminus \Theta_1$ to make sure that $\Theta_1 \cup \Theta_2$ are uniform in $[0, 1]$. Then, $\Theta_3 = [0.04, 0.44, 0.69, 0.98]$ was sampled to ensure the uniformity of $\bigcup_{i=2}^3 \Theta_i$.

The parameters in Θ_1, Θ_2 , and Θ_3 are represented as black, red, and green points in Figure 1, respectively. Each row in the figure represents the parameters at the corresponding time step. By using the reverse sequential sampling scheme, the training samples have good space filling in $[0, 1]$, even though the number of training parameters decreased when $t \in \Theta_3 \setminus \Theta_2 = \{11, 12, 13, 14, 15\}$.

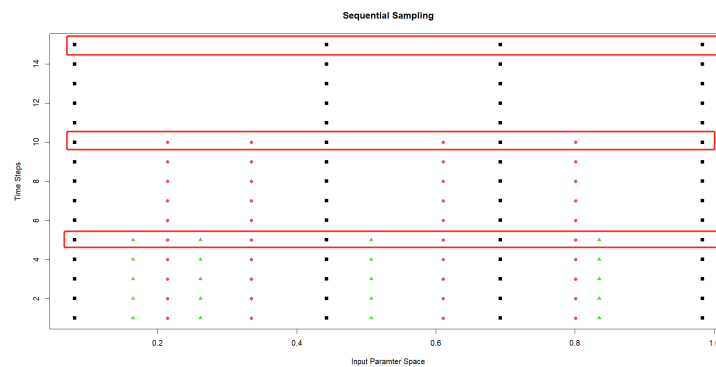


Figure 1. An illustration of the reverse sequential sampling scheme.

The collected dataset from all the $\sum_{i=1}^s c_i$ simulations is then used to train a surrogate model. Because there are many time steps and space nodes, building a surrogate model to respond at all temporal and spatial points is expensive. POD is a common method for reducing the dimensions of spatio-temporal data. For spatio-temporal data $f(x_n, t_m)$, x_n represents a spatial node, and t_m represents a time step. $\overline{f(x_n)}$ represents the average over the time steps of node x_n . Let $y(x_n, t_m)$ represent the spatio-temporal data minus the mean, i.e.,

$$y(x_n, t_m) = f(x_n, t_m) - \overline{f(x_n)}, \tag{3}$$

and the spatio-temporal data can be denoted as

$$X \triangleq \begin{bmatrix} y(x_1, t_1; \theta) & \cdots & y(x_1, t_M; \theta) \\ \vdots & \ddots & \vdots \\ y(x_N, t_1; \theta) & \cdots & y(x_N, t_M; \theta) \end{bmatrix}_{N \times M}. \tag{4}$$

Then, POD is used to decompose the real-valued $N \times N$ matrix XX^T by eigenvalue decomposition. Let $\Phi = [\phi_1, \phi_2, \dots, \phi_N]$, where $\phi_j \in \mathbb{R}^N$ is the standard orthogonal feature vector, and $\lambda_1, \lambda_2, \dots, \lambda_N$ is the corresponding eigenmatrix and eigenvalues, respectively. The number of modes K is chosen such that $\frac{\sum_{i=1}^K \lambda_i}{\sum_{i=1}^N \lambda_i} \geq 99\%$, where the proportion of eigen-

value $\frac{\lambda_j}{\sum_{i=1}^N \lambda_i}$ represents the energy contained in the corresponding eigenvector ϕ_j . When considering that the spatio-temporal data removed the mean, $f(x, t)$ can be decomposed as

$$f(x, t) \approx \sum_{k=1}^K \beta_k(t) \phi_k, \tag{5}$$

where $\beta_k(t_m)$ is the time-varying coefficient corresponding to the k -th mode at time step t_m , which can be calculated by

$$\beta_k(t_m) = \langle f(t_m), \phi_k \rangle, \tag{6}$$

where $f(t_m) = [f(x_1, t_m), f(x_2, t_m), \dots, f(x_N, t_m)]^T$. Consider the modes $\{\phi_1, \phi_2, \dots, \phi_K\}$ as a common basis function, and the spatio-temporal data

$$\begin{bmatrix} f(x_1, t_1; \theta_{11}) & \cdots & f(x_1, t_{M_1}; \theta_{11}) \\ \cdots & \cdots & \cdots \\ f(x_N, t_1; \theta_{11}) & \cdots & f(x_N, t_{M_1}; \theta_{11}) \end{bmatrix}_{N \times M_1}, \dots, \begin{bmatrix} f(x_1, t_1; \theta_{sc_s}) & \cdots & f(x_1, t_{M_1}; \theta_{sc_s}) \\ \cdots & \cdots & \cdots \\ f(x_N, t_1; \theta_{sc_s}) & \cdots & f(x_N, t_{M_1}; \theta_{sc_s}) \end{bmatrix}_{N \times M_s} \tag{7}$$

can be represented by

$$\begin{bmatrix} \beta_1(t_1; \theta_{11}) & \cdots & \beta_1(t_{M_1}; \theta_{11}) \\ \cdots & \cdots & \cdots \\ \beta_K(t_1; \theta_{11}) & \cdots & \beta_K(t_{M_1}; \theta_{11}) \end{bmatrix}_{K \times M_1}, \dots, \begin{bmatrix} \beta_1(t_1; \theta_{sc_s}) & \cdots & \beta_1(t_{M_1}; \theta_{sc_s}) \\ \cdots & \cdots & \cdots \\ \beta_K(t_1; \theta_{sc_s}) & \cdots & \beta_K(t_{M_1}; \theta_{sc_s}) \end{bmatrix}_{K \times M_s}. \tag{8}$$

Note that, through the POD, the $N \times M$ dimensional matrix is reduced to the $K \times M$ dimensional matrix, which means that the spatio-temporal data with N space points and M time points are reduced from $N \times M$ to $K \times M$, where $N \gg K$.

2.2. The Proposed Model

After the decomposition of POD, the surrogate model can be built for each β_k in (8). Since the establishment of the surrogate model is the same for every k , hereafter we omit k and let $\beta(t_m; \theta)$ denote $\beta_k(t_m; \theta)$. Section 2.2.1 will introduce the steps for building the surrogate model for $\beta(t_m; \theta), j = 1, 2, \dots, t_{M_s}$. A weighted model is proposed in Section 2.2.2 to improve the stability of the model, and in Section 2.2.3, an algorithm to predict the system dynamics under new parameters is presented.

2.2.1. The Surrogate Model for Time-Varying Coefficients

To simplify the elaboration, we first introduce the proposed method with only two time spans $\mathcal{T}_1 = \{t_1, t_2, \dots, t_{M_1}\}$ and $\mathcal{T}_2 = \{t_1, t_2, \dots, t_{M_1}, \dots, t_{M_2}\}$, where $t_{M_2} = t_M$. First, the c_2 and c_1 parameters are sequentially sampled from the parameter space Ω according to the algorithm proposed in the previous section and are denoted as Θ_2 and Θ_1 , respectively. Each set of parameters $\theta_{2j} \in \Theta_2, j = 1, 2, \dots, c_2$ is plugged into the simulator to obtain the spatio-temporal field data $y = f(x, t), x = x_1, x_2, \dots, x_N; t = t_1, t_2, \dots, t_{M_1}, t_{M_1+1}, \dots, t_{M_2}$. For each set of parameters $\theta_{1j} \in \Theta_1, j = 1, 2, \dots, c_1$, corresponding simulations are made over $[0, t_{M_1}]$. Then, we have

$$\begin{bmatrix} \beta(t_1; \theta_{11}) & \cdots & \beta(t_{M_1}; \theta_{11}) \\ \beta(t_1; \theta_{12}) & \cdots & \beta(t_{M_1}; \theta_{12}) \\ \cdots & \cdots & \cdots \\ \beta(t_1; \theta_{1c_1}) & \cdots & \beta(t_{M_1}; \theta_{1c_1}) \\ \beta(t_1; \theta_{21}) & \cdots & \beta(t_{M_1}; \theta_{21}) \\ \cdots & \cdots & \cdots \\ \beta(t_1; \theta_{2c_2}) & \cdots & \beta(t_{M_1}; \theta_{2c_2}) \end{bmatrix}_{(c_1+c_2) \times M_1} \quad \text{and} \quad \begin{bmatrix} \beta(t_{M_1+1}; \theta_{21}) & \cdots & \beta(t_{M_2}; \theta_{21}) \\ \beta(t_{M_1+1}; \theta_{22}) & \cdots & \beta(t_{M_2}; \theta_{22}) \\ \cdots & \cdots & \cdots \\ \beta(t_{M_1+1}; \theta_{2c_2}) & \cdots & \beta(t_{M_1}; \theta_{2c_2}) \end{bmatrix}_{c_2 \times (M_2-M_1)} \tag{9}$$

It can be seen from (9) that the training data for the surrogate model at each time step has a different size. For $t_m \in \{t_1, t_2, \dots, t_{M_1}\}$, there are $c_1 + c_2$ input parameters, so the kriging models are built for $\beta(t_m; \theta)$ [15–18]. For t_{M_1+1} , the number of input parameters is

reduced to c_2 , which is less than that of the previous time step t_{M_1} . When the number of training parameters is insufficient, the kriging model is not effective [15,16]. If the kriging model is built directly based on the spatio-temporal data for these parameters, the accuracy of the model will be reduced, which will further affect the prediction accuracy for the flow structures and dynamics of systems. Hence, we propose the cokriging method and the weighted method to solve this problem by using the time correlation.

Given the covariance function $C(\cdot) = \sigma^2 R(\theta, \theta'; \kappa)$, the kriging model can model the correspondence and uncertainty between the input parameters and the responses. For every t_m , the form of the ordinary kriging model with the response $\beta(t_m; \theta)$ is

$$\hat{\beta}(t_m; \theta) = b + Z_m(\theta), \quad Z_m(\theta) \sim N\left(0, \sigma^2 R(\theta, \theta'; \kappa)\right), \tag{10}$$

where $Z_m(\theta)$ is a zero mean stationary Gaussian process (GP) with mean zero, variance σ^2 and correlation function $R(\theta, \theta'; \kappa)$, b is the mean. In this paper, we utilize the Gaussian correlation function of the form

$$R(\theta, \theta'; \kappa) = \exp\left[-\sum_{r=1}^d \kappa_r (\theta_{(r)} - \theta'_{(r)})^2\right], \tag{11}$$

where $\kappa = [\kappa_1, \dots, \kappa_d]^T$, κ_r are the unknown correlation parameters used to fit the model, $\theta_{(r)}$ is the r -th dimension of θ [15,16]. Then, the unknown parameter of the ordinary kriging model is $(b, \sigma^2, \kappa)^T$, which can be estimated by the maximum likelihood empirical best linear unbiased prediction [15].

The cokriging models [19] will be employed to build the surrogate models for t_m , $m = M_1 + 1, M_1 + 2, \dots, M_2$ with respect to the input parameters Θ_2 by considering the time correlation between different time steps. The fundamental idea is to establish a relational model between the original data and highly correlated data such that the prediction capability of the surrogate model is enhanced. This scheme enables a more accurate predictive model to be built for the auxiliary data, which will help capture the trend of the response varying in parametric space. Gratiet and Garnier [20] improved the cokriging model by constructing a recursive computation scheme. The surrogate model for $\beta(t_{M_1+1}; \theta)$ can be formulated as

$$\begin{cases} \hat{\beta}(t_{M_1+1}; \theta) = z_{M_1+1}(\theta) = \rho_{M_1}(\theta)z_{M_1}(\theta) + \delta_{M_1+1}(\theta) \\ z_{M_1}(\theta) \perp \delta_{M_1+1}(\theta) \\ \rho_{M_1}(\theta) = g_{M_1}^T(\theta)\alpha_{\rho_{M_1}} \end{cases}, \tag{12}$$

where $\rho_{M_1}(\theta)$ is the adjustment function [20], z_{M_1} is a GP model for $\beta(t_{M_1}; \theta)$ which is given by (10), and $\delta_{m+1}(\theta)$ is a GP model for the difference of $\beta(t_{M_1+1}; \theta)$ and $\beta(t_{M_1}; \theta)$. \perp denotes the independence relationship between z_{M_1} and $\delta_{M_1+1}(\theta)$, and g_{M_1} is a vector of regression functions with its coefficient $\alpha_{\rho_{M_1}}$. $\rho_{M_1}(\theta)$ can be considered as a constant [19], which is used in this paper. Further details of the cokriging model are introduced in [19,20]. With (12), the predictions of $\beta(t_{m+1}; \theta)$ at $\theta \in \Theta_1$ are given and denoted as $\hat{\beta}(t_{m+1}; \theta)$. Then, $\beta(t_{M_1+1}; \theta_{2j}), j = 1, 2, \dots, c_2$ and $\hat{\beta}(t_{M_1+1}; \theta_{1j}), j = 1, 2, \dots, c_1$ are used to build the surrogate model for $\beta(t_{M_1+2}; \theta)$. The surrogate models are built sequentially by (12) for $t_{M_1+1}, t_{M_1+2}, \dots, t_{M_2}$. This sequential approach makes it easy to generalize scenarios for three or more time spans.

2.2.2. The Weighted Surrogate Model for Time-Varying Coefficients

The method in Section 2.2.2 can sequentially give predictions for $\beta(t_m; \theta)$ when $t = t_{M_1+1}, t_{M_1+2}, \dots, t_{M_2}$. However, the above method has the disadvantage of using the prediction of the previous step t_m as auxiliary data to establish a cokriging model at the next time step t_{m+1} , where $m = M_1, M_1 + 1, \dots, M_2 - 1$.

We propose a weighted method based on the Pearson correlation coefficient. As is suggested by [20], the correlation can be defined by the Pearson correlation coefficient

$$r_{t_m, t_{m'}} = \frac{\text{cov}(\beta(t_m; \theta), \beta(t_{m'}; \theta))}{\sigma_{\beta(t_m; \theta)} \sigma_{\beta(t_{m'}; \theta)}}. \tag{13}$$

The closer the $r_{t_m, t_{m'}}$ is to 1, the stronger the correlation between the two sets of data. For $m = M_1, M_1 + 1, \dots, M_2 - 1$, two surrogate models are built for $\beta(t_m; \theta)$: one is the cokriging model from (12) with $\beta(t_{M_1}; \theta)$ as the auxiliary data, denoted as $\hat{\beta}_{\text{Cok}}(t_m; \theta)$, and the other is the ordinary kriging from (10), denoted as $\hat{\beta}_{\text{Ok}}(t_m; \theta)$. The Pearson correlation coefficient $r = r_{t_m, t_{M_1}}$ is calculated as the weight. Then, the weighted surrogate model can be formulated as

$$\hat{\beta}_W(t_m; \theta) = \begin{cases} r \hat{\beta}_{\text{Cok}}(t_{m_1+1}; \theta) + (1 - r) \hat{\beta}_{\text{Ok}}(t_{m_1+1}; \theta), & r \geq r_0 \\ \hat{\beta}_{\text{Ok}}(t_{m_1+1}; \theta), & r < r_0 \end{cases}. \tag{14}$$

where the threshold r_0 is a pre-fixed constant.

2.2.3. The Prediction of System Dynamics with New Input Parameters

The kriging models for $\beta(t_m; \theta)$ are trained independently for $t \in \{t_1, t_2, \dots, t_{M_1}\}$, and for $t \in \{t_{M_1+1}, t_{M_1+2}, \dots, t_{M_2}\}$, the surrogate models for $\beta(t_m; \theta)$ are given by the cokriging method or weighted method according to (12) or (14), respectively.

Let θ^* be a new input parameter for which prediction is desired. Then, the prediction of the new system dynamics at \mathcal{T}_2 can be given by the reconstruction of the predicted time-varying coefficients $\hat{\beta}_k(t, \theta^*)$ and the given mode $\phi_k(x)$:

$$\hat{f}(x, t; \theta^*) = \sum_{k=1}^K \hat{\beta}_k(t, \theta^*) \cdot \phi_k(x). \tag{15}$$

Our whole spatio-temporal surrogate model framework is summarized as Algorithm 1. Before building the surrogate model, there are some settings related to the model that need to be determined.

- Determine the input parameters θ and their ranges and map the input parameter space Θ to $[0, 1]^p$, where p is the dimension of the input parameters.
- Determine the spatial extent and time span of the spatio-temporal data $f(x, t; \theta)$, and make discrete divisions of the spatio-temporal field with appropriate precision, which is denoted as $\mathcal{X} \times \mathcal{T}$.
- Determine the different ending times of the simulation of spatio-temporal data $t_{M_1}, t_{M_2}, \dots, t_{M_s}$ and the corresponding number of training samples n_1, n_2, \dots, n_s .

Algorithm 1: The framework for the spatio-temporal surrogate model.

- 1 Sample the training parameter sets $\Theta_s, \Theta_{s-1}, \dots, \Theta_2$, and Θ_1 through sequential sampling, and let $\Theta = \Theta_1 \cup \Theta_2 \cup \dots \cup \Theta_{s-1} \cup \Theta_s$.
 - 2 Generate the spatio-temporal data through the simulator. For the parameter in the i th parameter set Θ_i , the corresponding spatio-temporal data are calculated as $f(x, t; \theta), x \in \mathcal{X}, t \in \mathcal{T}_i = \{t_1, t_2, \dots, t_{M_i}\}, i = 1, 2, \dots, s$.
 - 3 Decompose the spatio-temporal data through POD, give the number of POD modes T and write it as $\phi_1, \phi_2, \dots, \phi_K$. Furthermore, calculate the corresponding time-varying coefficient $\beta_k(t; \theta)$ for the k -th mode at the instant of time t at parameter θ .
 - 4 **for** $1 \leq k \leq K$ **do**
 - 5 **for** $t_m \in \{t_1, t_2, \dots, t_{M_1}\}$ **do**
 - 6 \lfloor Establish the Ordinary Kriging model for $\beta_k(t; \theta)$.
 - 7 **for** $t_m \in \{t_{M_1+1}, t_{M_1+2}, \dots, t_M\}$ **do**
 - 8 \lfloor Establish the Weighted model or CoKriging model for $\beta_k(t; \theta)$.
 - 9 Do prediction. After building the surrogate model for each moment, give predictions $\hat{\beta}_k(t; \theta)$ for $\beta_k(t; \theta^*)$ under the new input parameters θ^* sequentially and reconstruct the new spatio-temporal data $\hat{f}(x, t; \theta^*) = \sum_{k=1}^K \hat{\beta}_k(t; \theta) \cdot \phi_k(x)$.
-

3. Simulation Studies

3.1. The Case of 2D Input Parameters for Spatio-Temporal Data

As a simple example, a one-dimensional advection equation is considered, with the initial phase φ_0 and wave speed v as the input parameter. The advection equation represents a wave propagating with a constant velocity [21,22] of the form

$$\frac{\partial \mu}{\partial t} + v \frac{\partial \mu}{\partial x} = 0, \quad x \in [-2, 2]. \tag{16}$$

The initial condition is

$$\mu = \mu_0(x, 0) = \begin{cases} 0, & |x| > 1 \\ A \sin(\omega x + \varphi_0), & |x| \leq 1 \end{cases} \tag{17}$$

and the boundary condition is

$$\mu(-2, t) = \mu(2, t). \tag{18}$$

Besides the input parameter $\theta = (\varphi_0, v)^T$, the above advection equation has two other control parameters, namely amplitude A and frequency ω . We fixed the frequency $\omega = 2\pi$ and amplitude $A = 1$, and the input parameter space $\Omega = [0, 1] \times [0, 1]$. We take the space interval as $\Delta x = 0.01$, and the time interval as $\Delta t = 0.01$. The time range from 0 to 1 s is considered. Then, the spatio-temporal field is a grid of 401 space points and 101 time points. Let the interested time spans be $\mathcal{T}_1 = \{0, 0.01, 0.02 \dots, 0.49\}$ and $\mathcal{T} = \mathcal{T}_2 = \{0, 0.01, 0.02 \dots, 0.99, 1\}$. Our two proposed methods in Section 2.2.2 and in Section 2.2.3 are compared with the ordinary kriging method.

According to the reverse sequential sampling method proposed in Section 2.1, eight parameters were first sampled to form Θ_2 , and then, the other eight parameters were sampled to form Θ_1 . The parameters in $\Theta = \Theta_1 \cup \Theta_2$ are listed in Table 1. The spatio-temporal data at \mathcal{T}_1 were simulated under the parameters in Θ , and the spatio-temporal data at $\mathcal{T}_2 \setminus \mathcal{T}_1$ were simulated only under the input parameters in set Θ_2 . The training data were reduced by POD and the first 10 modes were chosen, which contain 99.21% information of the training data. The time correlations of the time-varying coefficients of the first five modes are shown in Figure 2. Next, the surrogate models were built for $\beta_k(t_m, \theta)$, $m = 1, 2, \dots, 101$, and $k = 1, 2, \dots, 10$. We refer to the three schemes of establishing the

surrogate model using the weighted method, cokriging method, and ordinary kriging methods as Scheme 1, Scheme 2, and Scheme 3, respectively.

Table 1. The training parameters.

θ	φ_0	v	θ	φ_0	v
θ_{11}	0.133	0.760	θ_{21}	0.176	0.662
θ_{12}	0.259	0.555	θ_{22}	0.688	0.186
θ_{13}	0.782	0.268	θ_{23}	0.951	0.338
θ_{14}	0.564	0.143	θ_{24}	0.405	0.891
θ_{15}	0.460	0.417	θ_{25}	0.327	0.456
θ_{16}	0.641	0.014	θ_{26}	0.043	0.608
θ_{17}	0.081	0.920	θ_{27}	0.820	0.079
θ_{18}	0.878	0.729	θ_{28}	0.545	0.975

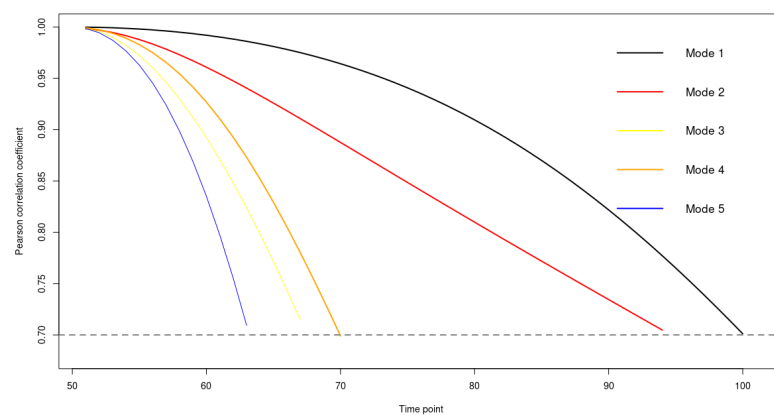


Figure 2. Time correlations of the time-varying coefficient corresponding to the first five POD modes in the advection equation example.

Scheme 1: For $\beta(t_m; \theta)$, $t_m \in \mathcal{T}_1$, there were $|\Theta_1| + |\Theta_2| = 16$ groups of training data, where $|\cdot|$ means the element number of the set. The ordinary kriging models were established for $\beta(t_m; \theta)$ separately. For $\beta(t_m; \theta)$, $t_m \in \mathcal{T}_2 \setminus \mathcal{T}_1$, there were only $|\Theta_2| = 8$ groups of training data. Furthermore, the weighted method, which is proposed in Section 2.2.3, was used for $t_m \in \mathcal{T}_2 \setminus \mathcal{T}_1$. r_0 is chosen as 0.7.

Scheme 2: For $\beta(t_m; \theta)$, $t_m \in \mathcal{T}_1$, the model is the same as Scheme 1, and for $\beta(t_m; \theta)$, $t_m \in \mathcal{T}_2 \setminus \mathcal{T}_1$, the cokriging method is used for $\beta_k(t, \theta)$ as introduced in Section 2.2.2.

Scheme 3: For $\beta(t_m; \theta)$, $t_m \in \mathcal{T}_2$, the ordinary kriging models are established separately.

The simulation results at $\theta_1^* = (0.483, 0.427)^T$ are chosen as the test data, which are used to evaluate the performance of the methods. For all three schemes, the predictions are given by (2). The predictions of the three methods for the spatio-temporal data under new parameters at $t_{51} = 0.5s$ and $t_{101} = 1.0s$ are shown in Figure 3. The predictions given by Scheme 1 and Scheme 2 are close to each other at t_{51} , and at t_{101} , and Scheme 1 is better than Scheme 2. Both Scheme 1 and Scheme 2 are significantly better than Scheme 3, which does not accurately capture the law of wave change. Figure 4 shows the prediction errors at each time step, defined as the average of the squared error of all spatial points. At t_{51} , the training parameters dropped from 16 to 8, which caused the prediction error of the ordinary kriging method to increase immediately, but both of our methods avoided the rapid increases because they used the information of the previous time steps. The weighted method includes the ordinary kriging method, which does not perform very well when the training data are reduced, but avoids predictions that deviate too far from the actual value when the data correlation weakens. In contrast with the cokriging method, the weighted method is therefore better in this example.

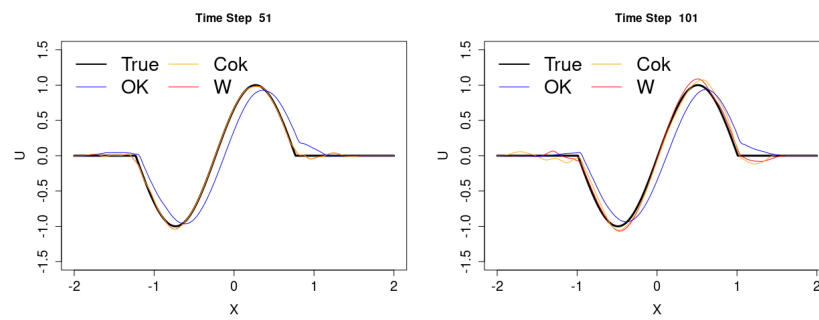


Figure 3. The left and right panels show the true and predicted waves at $t_{51} = 0.5$ s and $t_{101} = 1.0$ s, respectively. The black lines are the true data, and the red, orange, and blue lines are the predictions given by Schemes 1, 2, and 3, respectively.

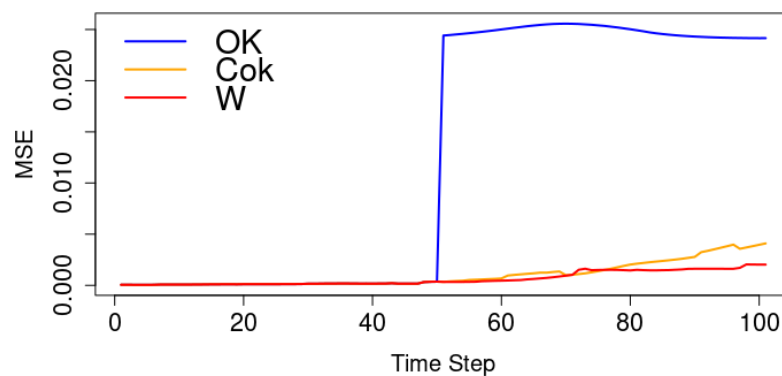


Figure 4. The prediction errors at each time step.

One simulation considers the effect of the number of input parameters, which is used to run further simulations. The details regarding the selection of parameters are shown in Table 2. For each case, we randomly generated the training set and built the model 100 times, and for every trained surrogate model, 20 test parameters were sampled to verify the prediction error of the model.

Table 2. The number of input parameters in the simulations.

Case	Number
Case 1	$c_1 = c_2 = 8$
Case 2	$c_1 = c_2 = 10$
Case 3	$c_1 = c_2 = 12$
Case 4	$c_1 = c_2 = 14$
Case 5	$c_1 = c_2 = 16$

The mean square error (MSE) is used as a metric to evaluate the performance of the method, which is defined as

$$MSE = \frac{1}{n_{te} \cdot S \cdot T} \sum_{i=1}^{n_{te}} \sum_{j=1}^S \sum_{k=1}^T [f(x_j, t_k; \theta_i) - \hat{f}(x_j, t_k; \theta_i)]^2. \tag{19}$$

Figure 5 shows the boxplot of the MSE of our two proposed methods, and for different numbers of training samples, the weighted method is consistently better than the cokriging method; Figure 6 shows that the mean MSE of the three schemes varies with the amount of data, from which we can see that the MSE of the method we proposed is much smaller than that of the ordinary kriging method.

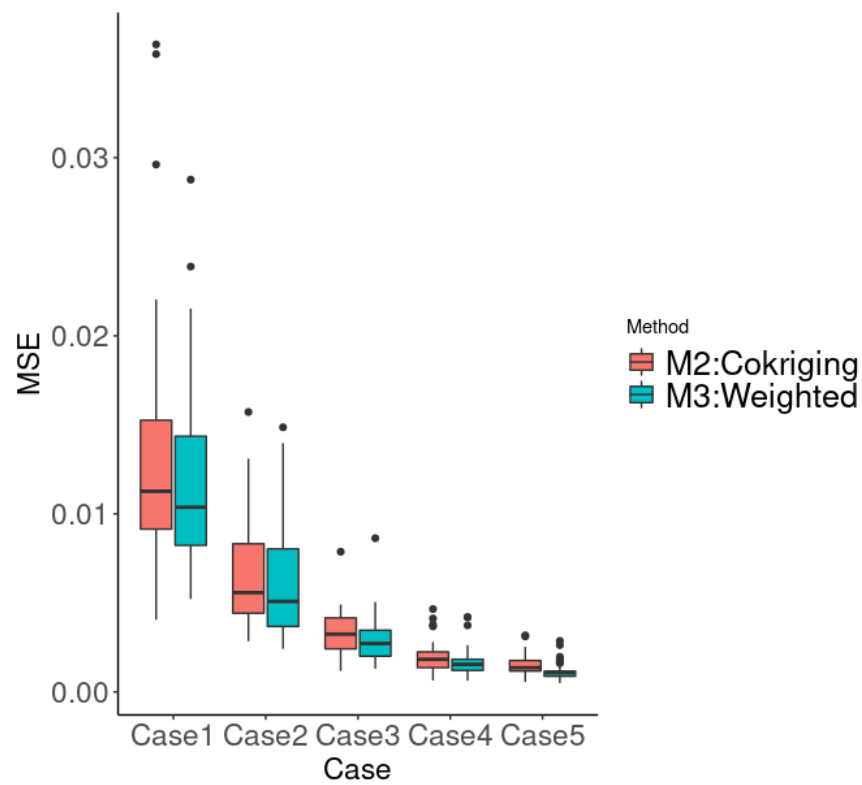


Figure 5. The MSE of our two proposed methods.

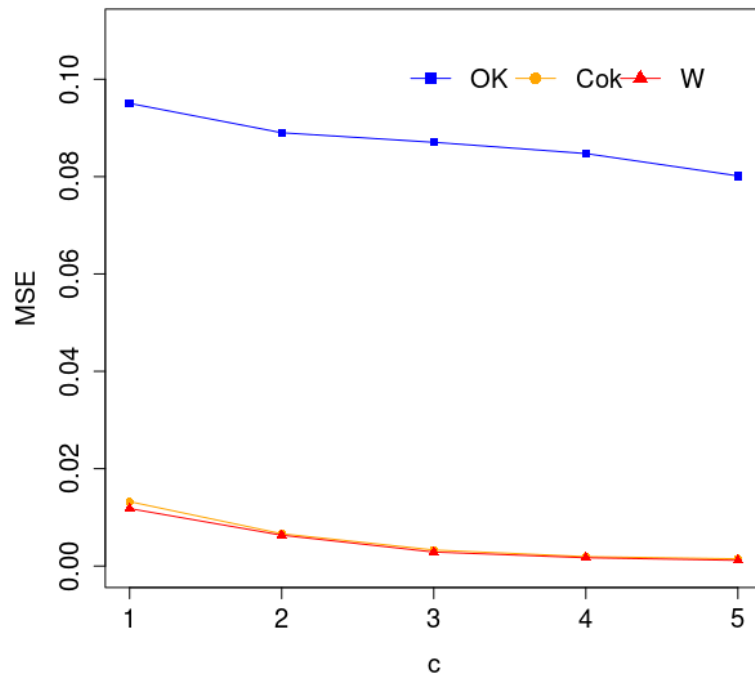


Figure 6. The mean MSE of the three schemes varies with the amount of data.

3.2. The Case of Canadian Weather

Our proposed methods are used for real observational spatio-temporal data. The R package *fda* provides Canadian weather data, including observations of daily temperature and precipitation at 35 different locations in Canada. Suppose that some of these stations have only the first 250 days of observations (represented by black squares in Figure 7), some

have only the first 300 days of observations (represented by red circles in Figure 7), and some have all the observations (represented by green triangles in Figure 7). Furthermore, suppose that there are four stations without observational data that thus require prediction; these are considered the test data.

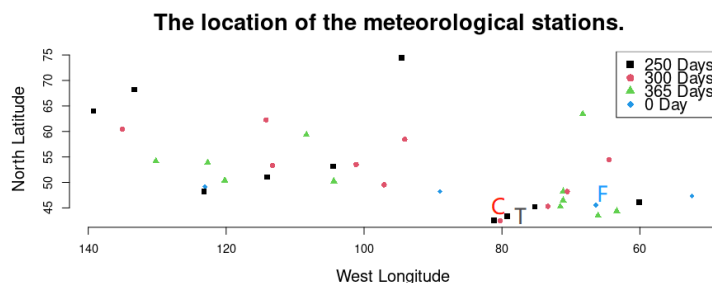


Figure 7. The location of 35 weather stations. C, T, and F represent the locations of Charlottetown, Toronto, and Fredericton, respectively.

The stations at Charlottetown and Toronto were missing data for 65 days and 115 days, respectively, and the station at Fredericton had no observational data. We care about the prediction error for the time period when the data are missing. As a comparison, we built models using three methods, the ordinary kriging method, our proposed cokriging method, and our proposed weighted method. Figure 8 shows the real and forecast temperatures by the three methods at three weather stations. The two methods we proposed are significantly better than the ordinary kriging method. The MSE of the weighted method, cokriging method, and ordinary kriging method for the temperature of days 301 to 365 at the Charlottetown station are 1.454, 3.071, and 79.107, respectively. The MSE of the weighted method, cokriging method, and ordinary kriging method for the temperature of days 251 to 365 at the Toronto station are 3.263, 1.982, and 52.713, respectively. The MSE of the weighted method, cokriging method, and ordinary kriging method for the annual temperature at the Fredericton station are 3.361, 4.101, and 4.496, respectively.

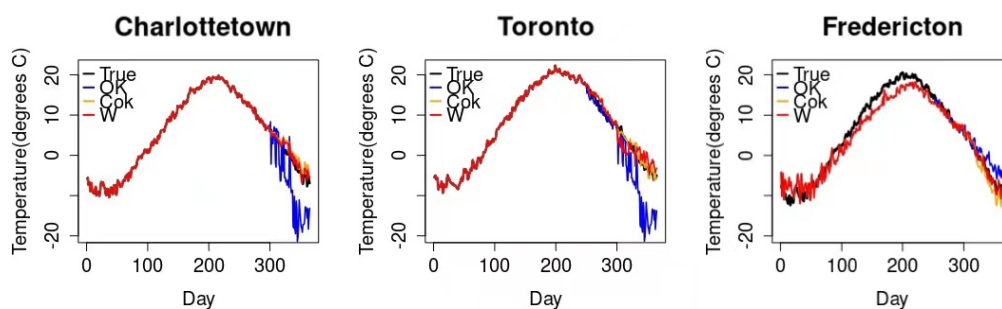


Figure 8. Predictions of Canadian weather.

In the example of the advection equation, the time correlation under some modes decreases very quickly—see Figure 2. In this situation, the weighted method performs better than the cokriging method. In the example of Canadian weather, the time correlation falls off more slowly relative to the advection equation example (Figure 9), and the cokriging method can also be used.

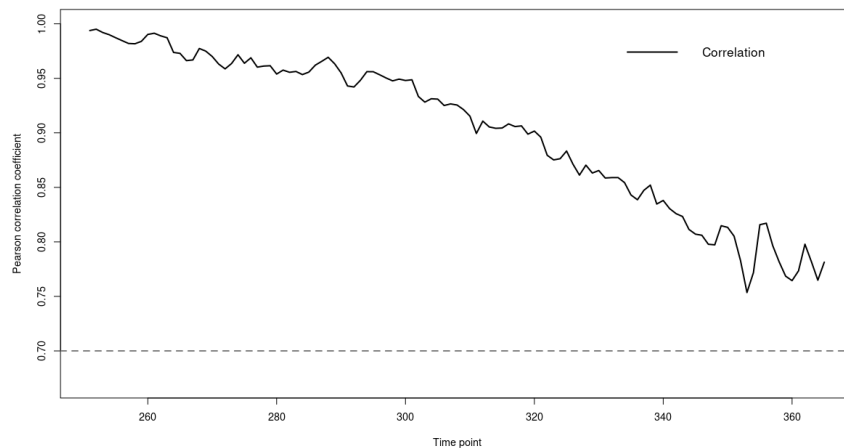


Figure 9. Time correlations of temperature in the Canadian weather example.

4. Application of the Model to Bai River Data

In this section, real data analysis for Bai River, which is located in Sichuan Province, as shown in Figure 10, is used to illustrate the performance of the proposed method. In the upper reaches of the river, there are some factory sewage pipes discharging a certain pollutant into the river. Assuming that there are two sewage outlets, A and B, then there are four-dimensional input parameters, the pollutant concentrations and water flow velocities of the two outlets, which are denoted as $\theta = (p_A, p_B, v_A, v_B)^T \in [0, 1]^4$.

The simulator is the C++ code based on [23], which divides the geographic space into 37,960 mesh points. The time interval was $\Delta t = 0.1s$, and the solver was stored every 1000 steps. The solver with $M_2 = 1950$ time steps was calculated, which simulated the pollutant concentration in the Bai River over a 54 h period.

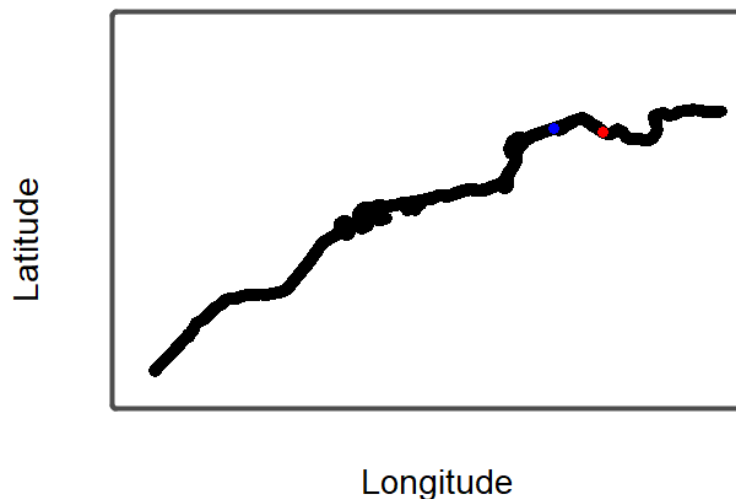


Figure 10. The shape of the Bai River.

According to the reverse sequential sampling scheme, 10 parameters are selected for Θ_2 , and another 10 parameters are selected for Θ_1 , which is used to run the simulations. Suppose that when the simulation reached $t_{M_1}, M_1 = 1000$, the detectors downstream of the river identified the high-level warning line of pollutants in the river. Thus, we needed to predict the change in pollutants in the future. However, we did not have enough time to simulate the spatio-temporal data under all parameters. Only the spatio-temporal data corresponding to the parameters in Θ_2 were simulated, and a surrogate

model was established in order to further judge which sewage outlet discharged pollutants excessively and caused the increase in pollution. The weighted method, cokriging method, and ordinary kriging method are used to establish the surrogate models. Figure 11 shows how pollutants at the red point and blue point in the Bai River map (Figure 10) changed over time. Figure 12 shows the prediction error of the three methods over the whole time span, which shows that two new methods are better than the original method.

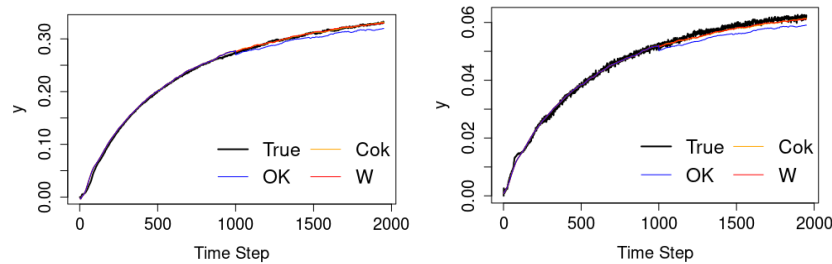


Figure 11. Changes in pollutants at one point in the Bai River over the whole time span.

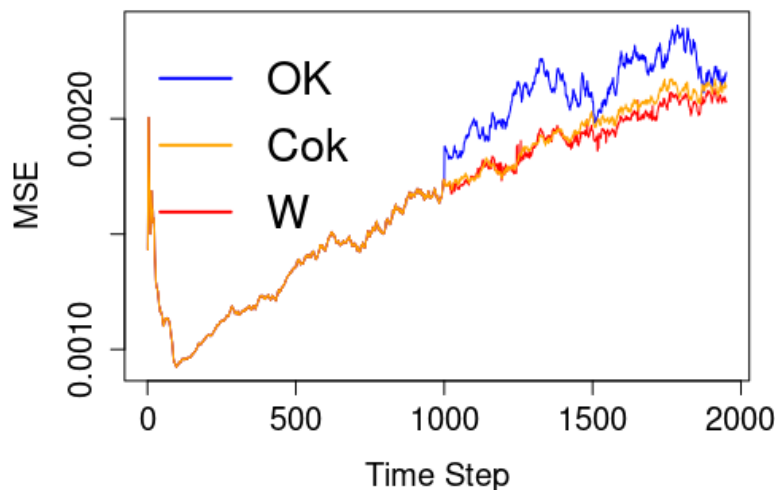


Figure 12. The MSE at each time step.

5. Conclusions

Simulation is a common approach to the investigation of complex phenomena and systems. However, simulations are very expensive due to the requirement of solving large PDEs. How to best build surrogate models for simulations is an area still facing significant challenges. The goal of the surrogate models is to drastically reduce computational costs, especially when many predictions of spatio-temporal dynamics for unsimulated inputs are required. Most existing works in the literature only consider the situation with the same simulation time span. In this work, we consider the situation with multiple simulation time spans and propose a novel method to build efficient surrogate models. Firstly, a reverse sequential sampling method is presented to choose the input parameters for different simulation time spans. Then, a weighted surrogate model is proposed to fuse the spatio-temporal data from simulations with different time spans. The results of simulation studies and real data analysis based on Bai River in Sichuan Province of China show that the newly proposed method performs well and is superior to the traditional method.

The methods proposed in this work require that the spatio-temporal data are relatively smooth with respect to inputs and have correlations between time steps. The simulation studies also demonstrate that when the time correlations fall off quickly, the performance of the cokriging model deteriorates dramatically. Some random effects models could

be considered to deal with this challenge, and will be studied in future work. The non-stationary nature of the data is another new challenge that will be considered.

Author Contributions: Conceptualization, D.W. and Y.T.; writing—original draft preparation, Y.H.; writing—review and editing, D.W. and Y.T.; funding acquisition, D.W. and Y.T. All authors have read and agreed to the published version of the manuscript.

Funding: Yue Huan is supported by the China Scholarship Council. Yue Huan and Dianpeng Wang are supported by the National Natural Science Foundation of China (grant no. NSFC 11801034 and grant no. NSFC 12171033) and the Fundamental Research Funds for Central Public Welfare Scientific Research Institutes of China (2019YSKY-019), and Yubin Tian is supported by the National Natural Science Foundation of China (grant no. NSFC 12131001).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

POD	Proper Orthogonal Decomposition
ROM	Reduced Order Modeling
LHD	Latin Hypercube Design
GP	Gaussian Process
OK	Ordinary Kriging
MSE	Mean Square Error

References

- Xiao, C.; Leeuwenburgh, O.; Lin, H.X.; Heemink, A. Non-intrusive subdomain POD-TPWL for reservoir history matching. *Comput. Geosci.* **2019**, *23*, 537–565. doi: 10.1007/s10596-018-9803-z. [CrossRef]
- Lumley, J.L. The structure of inhomogeneous turbulent flows. In *Atmospheric Turbulence and Radio Wave Propagation*; NAUKA: Moscow, Russia, 1967; pp. 166–178.
- Ioannidis, V.N.; Romero, D.; Giannakis, G.B. Inference of Spatio-Temporal Functions over Graphs via Multikernel Kriged Kalman Filtering. *IEEE Trans. Signal Process.* **2018**, *66*, 3228–3239. doi: 10.1109/TSP.2018.2827328. [CrossRef]
- Nguyen, H.; Katzfuss, M.; Cressie, N.; Braverman, A. Spatio-temporal data fusion for very large remote sensing datasets. *Technometrics* **2014**, *56*, 174–185. doi: 10.1080/00401706.2013.831774. [CrossRef]
- Shi, Q.; Dai, W.; Santerre, R.; Liu, N. A Modified Spatiotemporal Mixed-Effects Model for Interpolating Missing Values in Spatiotemporal Observation Data Series. *Math. Probl. Eng.* **2020**, *2020*, 1070831. doi: 10.1155/2020/1070831. [CrossRef]
- Guo, M.; Hesthaven, J.S. Data-driven reduced order modeling for time-dependent problems. *Comput. Methods Appl. Mech. Eng.* **2019**, *345*, 75–99. [CrossRef]
- Yeh, S.T.; Wang, X.; Sung, C.L.; Mak, S.; Chang, Y.H.; Zhang, L.; Wu, C.F.; Yang, V. Common proper orthogonal decomposition-based spatiotemporal emulator for design exploration. *AIAA J.* **2018**, *56*, 2429–2442. doi: 10.2514/1.J056640. [CrossRef]
- Chang, Y.H.; Zhang, L.; Wang, X.; Yeh, S.T.; Mak, S.; Sung, C.L.; Jeff Wu, C.F.; Yang, V. Kernel-smoothed proper orthogonal decomposition-based emulation for spatiotemporally evolving flow dynamics prediction. *AIAA J.* **2019**, *57*, 5269–5280. doi: 10.2514/1.J057803. [CrossRef]
- Chang, Y.H.; Wang, X.; Zhang, L.; Li, Y.; Mak, S.; Wu, C.F.J.; Yang, V. Reduced-Order Modeling for Complex Flow Emulation by Common Kernel-Smoothed Proper Orthogonal Decomposition. *AIAA J.* **2021**, *59*, 3291–3303. doi: 10.2514/1.j060574. [CrossRef]
- Chatterjee, A. An introduction to the proper orthogonal decomposition. *Curr. Sci.* **2000**, *78*, 808–817.
- Xiong, F.; Xiong, Y.; Chen, W.; Yang, S. Optimizing Latin hypercube design for sequential sampling of computer experiments. *Eng. Optim.* **2009**, *41*, 793–810. [CrossRef]
- Garud, S.S.; Karimi, I.A.; Kraft, M. Design of computer experiments: A review. *Comput. Chem. Eng.* **2017**, *106*, 71–95. [CrossRef]
- Joseph, V.R.; Gul, E.; Ba, S. Maximum projection designs for computer experiments. *Biometrika* **2015**, *102*, 371–380. [CrossRef]
- Carnell, R.; Carnell, M.R. Package ‘lhs’. CRAN. Available online: <http://cran.stat.auckland.ac.nz/web/packages/lhs/lhs.pdf> (accessed on 22 March 2022).
- Santner, T.J.; Williams, B.J.; Notz, W.I.; Williams, B.J. *The Design and Analysis of Computer Experiments*; Springer: New York, NY, USA, 2003; Volume 1.

16. Simpson, T.W.; Mauery, T.M.; Korte, J.J.; Mistree, F. Kriging models for global approximation in simulation-based multidisciplinary design optimization. *AIAA J.* **2001**, *39*, 2233–2241. [[CrossRef](#)]
17. Kleijnen, J.P. Kriging metamodeling in simulation: A review. *Eur. J. Oper. Res.* **2009**, *192*, 707–716. [[CrossRef](#)]
18. Vicario, G.; Craparotta, G.; Pistone, G. Meta-models in computer experiments: Kriging versus artificial neural networks. *Qual. Reliab. Eng. Int.* **2016**, *32*, 2055–2065. [[CrossRef](#)]
19. Kennedy, M.C.; O’Hagan, A. Predicting the output from a complex computer code when fast approximations are available. *Biometrika* **2000**, *87*, 1–13. doi: 10.1093/biomet/87.1.1. [[CrossRef](#)]
20. Gratiet, L.L.; Garnier, J. Recursive co-kriging model for design of computer experiments with multiple levels of fidelity. *Int. J. Uncertain. Quantif.* **2014**, *4*, 365–386. doi: 10.1615/int.j.uncertaintyquantification.2014006914. [[CrossRef](#)]
21. Hu, F.Q.; Hussaini, M.Y.; Rasetarinera, P. An analysis of the discontinuous Galerkin method for wave propagation problems. *J. Comput. Phys.* **1999**, *151*, 921–946. [[CrossRef](#)]
22. Vadyala, S.R.; Betgeri, S.N.; Betgeri, N.P. Physics-informed neural network method for solving one-dimensional advection equation using PyTorch. *Array* **2022**, *13*, 100110. [[CrossRef](#)]
23. Shen, H.; Parsani, M. Positivity-preserving CE/SE schemes for solving the compressible Euler and Navier–Stokes equations on hybrid unstructured meshes. *Comput. Phys. Commun.* **2018**, *232*, 165–176. [[CrossRef](#)]