



A Reproducing Kernel Hilbert Space Approach to Functional Calibration of Computer Models

Rui Tuo, Shiyuan He, Arash Pourhabib, Yu Ding & Jianhua Z. Huang

To cite this article: Rui Tuo, Shiyuan He, Arash Pourhabib, Yu Ding & Jianhua Z. Huang (2021): A Reproducing Kernel Hilbert Space Approach to Functional Calibration of Computer Models, *Journal of the American Statistical Association*, DOI: [10.1080/01621459.2021.1956938](https://doi.org/10.1080/01621459.2021.1956938)

To link to this article: <https://doi.org/10.1080/01621459.2021.1956938>



[View supplementary material](#) 



Published online: 01 Sep 2021.



[Submit your article to this journal](#) 



Article views: 467



[View related articles](#) 



CrossMark

[View Crossmark data](#) 

A Reproducing Kernel Hilbert Space Approach to Functional Calibration of Computer Models

Rui Tuo^a, Shiyuan He^b , Arash Pourhabib^c , Yu Ding^d, and Jianhua Z. Huang^e

^aIndustrial and Systems Engineering, Texas A&M University, College Station, TX; ^bInstitute of Statistics and Big Data, Renmin University of China, Beijing, China; ^cSchool of Industrial Engineering and Management, Oklahoma State University, Stillwater, OK; ^dTexas A&M, Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX; ^eDepartment of Statistics, Texas A&M University, College Station, TX

ABSTRACT

This article develops a frequentist solution to the functional calibration problem, where the value of a calibration parameter in a computer model is allowed to vary with the value of control variables in the physical system. The need of functional calibration is motivated by engineering applications where using a constant calibration parameter results in a significant mismatch between outputs from the computer model and the physical experiment. Reproducing kernel Hilbert spaces (RKHS) are used to model the optimal calibration function, defined as the functional relationship between the calibration parameter and control variables that gives the best prediction. This optimal calibration function is estimated through penalized least squares with an RKHS-norm penalty and using physical data. An uncertainty quantification procedure is also developed for such estimates. Theoretical guarantees of the proposed method are provided in terms of prediction consistency and consistency of estimating the optimal calibration function. The proposed method is tested using both real and synthetic data and exhibits more robust performance in prediction and uncertainty quantification than the existing parametric functional calibration method and a state-of-art Bayesian method.

ARTICLE HISTORY

Received July 2020
Accepted July 2021

KEYWORDS

Calibration; Computer experiment; Reproducing kernel Hilbert spaces; Smoothing splines; Uncertainty quantification

1. Introduction

To understand a physical system, one can conduct physical experiments by feeding a set of inputs to the system and observing the output. These inputs are called *control variables* to the system. The hope is that by learning the input–output relation of the system, in the future one can predict the output for any set of inputs that may be fed into the system. Because conducting physical experiments is usually very costly and inconvenient, computer simulation or using computer models is a common practice (Santner, Williams, and Notz 2003). A computer model attempts to use a set of mathematical formulas to mimic the input–output relation of the physical system and can be implemented through computer codes. There are usually a set of parameters in the computer model that represent intrinsic properties of the physical system. Different from control variables that one can determine and measure before an experiment, the computer model parameters are unobservable or unmeasurable. To determine the value of these parameters, one first obtains some data in the form of input–output pairs from physical experiments, and then adjusts the values of computer model parameters so that the computer model generates similar outputs for a given set of inputs in the physical experiments. This process is called *calibration*, and we refer to the computer model parameters as *calibration parameters*. Kennedy and O'Hagan (2001) is a renowned work on statistical framework for calibration of computer models, which also contains a review of earlier

work on the subject. Related references in this field include Higdon et al. (2004, 2008), Higdon et al. (2013), Bayarri et al. (2007a), Bayarri et al. (2007b), Tuo and Wu (2015), Tuo and Wu (2016), Joseph and Melkote (2009), Joseph and Yan (2015), Wong, Storlie, and Lee (2017), Plumlee (2017), Gu and Wang (2018), Plumlee (2019), Tuo (2019), Xie and Xu (2020), Wang et al. (2020), and the references therein.

Commonly used approaches to calibration are based on the assumption that there is one constant value for the calibration parameter, and the goal of calibration is to estimate that value, or find the closest possible point (Han, Santner, and Rawlinson 2009; Tuo and Wu 2016). However, in many complex physical systems, there may exist a functional relationship between calibration parameters and control variables. For example, in the resistance spot welding process discussed in Bayarri et al. (2007), or in the poly-vinyl alcohol (PVA)-treated buckypaper fabrication process (Pourhabib et al. 2015), engineering knowledge suggests that calibration parameters, that is, contact resistance in the former and the PVA absorption rate in the latter, are not fixed but a function of control variables. In these examples, any attempts that seek to calibrate the computer model by finding the “best” constant value for the calibration parameter will result in a significant mismatch between the outputs from the computer model and the physical experiment. Therefore, appropriate calibration of the computer model in such cases should allow the values of the calibration parameters to vary with the value of the

input variables. To highlight the functional relationship, we refer to the calibration parameters as the *functional calibration parameters* or *calibration functions*. If the calibration function has a known form with an unknown Euclidean parameter, one can develop a parametric solution to the functional calibration problem (Bayarri et al. 2007; Pourhabib et al. 2015; Atamturktur et al. 2015), and the theoretical framework of Tuo and Wu (2015, 2016) can be employed to find the optimal calibration parameter. However, such parametric solutions have obvious limitations, since the form of the calibration function is usually unavailable.

The goal of this article is to provide a theoretical framework for a nonparametric solution to the functional calibration problem and to develop corresponding computational methods and uncertainty quantification procedures. We formulate the problem such that it allows the data to determine the functional relationship between the calibration parameter and control variables, instead of imposing a parametric form for the relationship a priori. We use reproducing kernel Hilbert spaces to model the functional relationship and employ the penalized least squares for estimation. We devise a Gauss–Newton type of algorithm for computation and establish frequentist properties of our estimation procedure. We also develop an uncertainty quantification procedure by borrowing ideas from the smoothing splines literature. Our framework treats the computer model as an approximation to the physical system and do not assume there exists a “true” underlying calibration function. The optimal calibration function is defined operationally through an optimization problem to give the best prediction. Interestingly, when there are multiple functional calibration parameters, there is a possibility that the optimal calibration functions are not uniquely defined (see the discussion in Section 2.2), but our procedure still yields consistent prediction (Section 3.1) and our numerical results show good performance of our method in prediction (Section 6).

Upon completion of an earlier draft of this work, we became aware of two related articles that tackled the same problem using the Bayesian framework. Plumlee, Joseph, and Yang (2016) provided a Bayesian solution to the functional calibration problem in a specific application. Brown and Atamturktur (2018) developed a general Bayesian framework with a software implementation. Both articles utilized Gaussian process (GP) priors on the unknown calibration functions and applied the Markov chain Monte Carlo simulation for computation of the posterior distribution. While theoretical justification is lacking for these existing Bayesian functional calibration methods, we are able to establish an asymptotic theory of consistency and rates of convergence to provide some theoretical guarantees to our frequentist method. Our method still provides good prediction and uncertainty quantification when the optimal functional calibration parameter is not uniquely defined (see Tables 3 and 4). The existing Bayesian methods have not considered this challenging situation. This article also contains a comparative simulation study that is more comprehensive than those in the published works. Empirical evidence shows that the proposed frequentist approach tends to outperform existing Bayesian methods on the examples considered.

The rest of this article is organized as follows. Section 2 provides basic concepts regarding calibration, the formulation

of the functional calibration problem, and a solution procedure through penalized least squares. Section 3 provides theoretical properties of the proposed method. Section 4 presents a computational algorithm for solving the penalized least-square problem and derives the GCV criterion for penalty parameter selection. Section 5 develops an uncertainty quantification procedure for estimation of calibration function and for prediction. Section 6 presents the results for a simulation study. We illustrate the proposed method on real data in Section 7. Section 8 concludes the paper. Proofs of all theoretical results are given in supplementary materials.

Notations. For a vector \mathbf{a} , let $|\mathbf{a}|$ denote its Euclidean norm. For a function $f(\mathbf{x})$ defined on a domain \mathcal{X} , let $\|f\|_{L_2(\mathcal{X})} = \{\int_{\mathcal{X}} f^2(\mathbf{x}) d\mathbf{x}\}^{1/2}$ denote its L_2 norm. For a vector of functions $\mathbf{f} = (f_1, \dots, f_r)^T$, denote $\|\mathbf{f}\|_{L_2} = \{\sum_{i=1}^r \|f_i\|_{L_2(\mathcal{X})}^2\}^{1/2}$.

2. Functional Calibration: Problem Formulation

Consider a physical system that gives a vector of deterministic responses $\zeta(\mathbf{x}) \in \mathbf{R}^r$ when there is a vector of inputs (called control variables), $\mathbf{x} \in \mathcal{X}$, where \mathcal{X} is a convex and compact subset of \mathbf{R}^d . To learn the response function $\zeta(\cdot)$, we conduct physical experiments at design points $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathcal{X}$ and observe the corresponding response vectors, denoted as $\mathbf{y}_1^p, \dots, \mathbf{y}_n^p$, where superscript p stands for “physical.” A functional relationship exists between the input \mathbf{x} and the response $\zeta(\mathbf{x})$, but due to measurement noises we only observe a noisy version of the response. We assume that

$$\mathbf{y}_i^p = \zeta(\mathbf{x}_i) + \mathbf{e}_i, \quad i = 1, \dots, n, \quad (1)$$

where \mathbf{e}_i ’s are iid zero-mean random vectors. We call each pair $(\mathbf{x}_i, \mathbf{y}_i^p)$ a data point and the set $\{(\mathbf{x}_i, \mathbf{y}_i^p), i = 1, \dots, n\}$ the physical dataset.

2.1. Optimal Constant Calibration

Exploring the response function through mere physical experimentation can be extremely costly and time consuming. As such, a computer model is often utilized to simulate the physical system. The computer model takes $(\mathbf{x}, \boldsymbol{\theta})$ as the input and yields the computer model response $\mathbf{y}^s(\mathbf{x}, \boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \Theta$ denotes a calibration parameter, Θ is a subset of \mathbf{R}^q , and superscript s stands for “simulated,” referring to simulated from the computer model. The computer model response $\mathbf{y}^s(\mathbf{x}, \boldsymbol{\theta})$ is computable by running computer codes when a calibration parameter $\boldsymbol{\theta}$ and an input \mathbf{x} are given. An appealing feature of using computer models is that one can run the computer codes for the computer model for any feasible combination of control variables and calibration parameters, at much lower expense than running physical experiments. A fundamental problem for using computer models is *calibration*, that is, the problem of finding a suitable value for the calibration parameters so that the computer outputs match well those from the physical experiment for the same inputs or values of control variables.

Being built based on simplifying assumptions about the physical system, a computer model may not perfectly match the physical system. Therefore, we should not expect there exists a value $\boldsymbol{\theta}$ such that the computer model response $\mathbf{y}^s(\mathbf{x}, \boldsymbol{\theta})$ exactly

equals the physical system response $\zeta(\mathbf{x})$. The objective in calibration is to adjust the imperfect computer model, through changing the values of the calibration parameter, so that it adequately represents the physical system. Following Tuo and Wu (2015, 2016) and Wong, Storlie, and Lee (2017), we define the optimal value $\boldsymbol{\theta}^*$ of the calibration parameter $\boldsymbol{\theta}$ to be the value that minimizes the L_2 distance between $\zeta(\mathbf{x})$ and $\mathbf{y}^s(\cdot, \boldsymbol{\theta})$, that is,

$$\begin{aligned}\boldsymbol{\theta}^* &= \operatorname{argmin}_{\boldsymbol{\theta} \in \Theta} \|\zeta(\mathbf{x}) - \mathbf{y}^s(\mathbf{x}, \boldsymbol{\theta})\|_{L_2(\mathcal{X})}^2 \\ &= \operatorname{argmin}_{\boldsymbol{\theta} \in \Theta} \int_{\mathcal{X}} |\zeta(\mathbf{x}) - \mathbf{y}^s(\mathbf{x}, \boldsymbol{\theta})|^2 d\mathbf{x}.\end{aligned}\quad (2)$$

Empirically, the optimal calibration parameter can be estimated by solving a similar optimization problem as Equation (2) where the integral for the L_2 distance is replaced by sample average for a given physical dataset, and each $\zeta(\mathbf{x}_i)$ by its noisy observation \mathbf{y}_i^p . This empirical calibration method is called the ordinary least-square method in (Tuo and Wu 2015, sec. 4).

2.2. Optimal Functional Calibration

In the above discussion, we look for one single value of the calibration parameter such that the computer model best represents the physical experiment. We now consider the situation that there is a functional relationship between the calibration parameters and control variables. Our goal is to learn this functional relationship using physical data. Extending (2), we define the *optimal calibration function* for the functional calibration problem as

$$\begin{aligned}\boldsymbol{\theta}^*(\cdot) &= \operatorname{argmin}_{\boldsymbol{\theta}(\cdot)} \int_{\mathcal{X}} |\zeta(\mathbf{x}) - \mathbf{y}^s(\mathbf{x}, \boldsymbol{\theta}(\mathbf{x}))|^2 d\mathbf{x}, \\ \text{subject to: } \boldsymbol{\theta}(\mathbf{x}) &\in \Theta \text{ for all } \mathbf{x} \in \mathcal{X}.\end{aligned}\quad (3)$$

We refer to $\boldsymbol{\theta}^*(\mathbf{x})$ as the *optimal functional calibration parameter* and $\zeta^*(\mathbf{x}) = \mathbf{y}^s(\mathbf{x}, \boldsymbol{\theta}^*(\mathbf{x}))$ as the *optimal prediction* at location \mathbf{x} . Similar to Equation (2), here we do not assume a true value/form for the calibration function, but we seek a function that minimizes the discrepancy between the outputs of computer model and physical experiments. In fact, $\boldsymbol{\theta}^*(\cdot)$ can be defined pointwise: $\boldsymbol{\theta}^*(\mathbf{x})$ is the minimizer of the function $h_{\mathbf{x}}(\boldsymbol{\theta}) = |\zeta(\mathbf{x}) - \mathbf{y}^s(\mathbf{x}, \boldsymbol{\theta})|^2$ for each \mathbf{x} . Since Θ is compact, by assuming that $h_{\mathbf{x}}(\boldsymbol{\theta})$ is a continuous function, the minimizer $\boldsymbol{\theta}^*(\mathbf{x})$ always exists.

The optimal functional calibration parameter $\boldsymbol{\theta}^*(\mathbf{x})$ may not be unique, although the optimal prediction $\zeta^*(\mathbf{x})$ is uniquely defined. Section 6 presents some examples when there are multiple or even continuum many minimizers $\boldsymbol{\theta}^*(\mathbf{x})$ for (3); see simulation settings 3 and 4. A key condition for the uniqueness of $\boldsymbol{\theta}^*(\mathbf{x})$ is $r \geq q$, that is, the number of response variables is no fewer than the number of functional calibration parameters. To see this, note that $\boldsymbol{\theta} = \boldsymbol{\theta}^*(\mathbf{x})$ solves the system of r equations $\zeta^*(\mathbf{x}) = \mathbf{y}^s(\mathbf{x}, \boldsymbol{\theta})$, and thus in order for the system to have a unique solution, the number of equations should be no fewer than the number of parameters. Under mild conditions, the existence and uniqueness of the solution is ensured by the inverse function theorem or the implicit function theorem. On the other hand, since the primary goal of computer experiment is usually on prediction of outcomes of physical experiments, the uniqueness of $\boldsymbol{\theta}^*(\mathbf{x})$ is not that a big concern.

2.3. Empirical Functional Calibration

Recall that what we observe are noise-contaminated physical responses \mathbf{y}_i^p at a finite number of values of control variables $\mathbf{x}_i \in \mathcal{D}$. We need to learn the optimal calibration function using the physical data $\{\mathbf{x}_i, \mathbf{y}_i^p\}$. One could replace the integral in Equation (3) by a summation over the design points \mathbf{x}_i 's, replace each $\zeta(\mathbf{x}_i)$ by its noisy observation \mathbf{y}_i^p , and solve the corresponding minimization problem. However, this minimization problem does not have a unique solution since one can vary the values of the function at points other than the design points and do not change the value of the minimizing objective function. Therefore, to make $\boldsymbol{\theta}^*(\cdot)$ estimable, we have to postulate certain assumptions on $\boldsymbol{\theta}^*(\cdot)$. A common assumption is to suppose $\boldsymbol{\theta}^*$ has certain degree of smoothness. Therefore, we shall restrict our attention to elements in a suitably chosen space of smooth functions. Specifically, we consider *native spaces*, which is a generalization of *reproducing kernel Hilbert spaces* (RKHS). We refer to Wendland (2005) for the necessary mathematical background of RKHS and native spaces.

Let Φ be a conditionally positive-definite function over $\mathcal{X} \times \mathcal{X}$ and $\mathcal{N}_\Phi(\mathcal{X})$ be the native space generated by Φ with its native semi-norm $\|\cdot\|_{\mathcal{N}_\Phi}$. In general, the native semi-norm of a function f , $\|f\|_{\mathcal{N}_\Phi}$ is a measure of roughness of f , with a larger value indicating a rougher function. We define our empirical calibration function $\hat{\boldsymbol{\theta}}(\cdot)$ to be

$$\begin{aligned}\hat{\boldsymbol{\theta}}(\cdot) &:= \operatorname{argmin}_{\boldsymbol{\theta}(\cdot) = (\theta_1(\cdot), \dots, \theta_q(\cdot))^T} \frac{1}{n} \sum_{i=1}^n \left| \mathbf{y}_i^p - \mathbf{y}^s(\mathbf{x}_i, \boldsymbol{\theta}(\mathbf{x}_i)) \right|^2 \\ &\quad + \lambda \sum_{j=1}^q \|\theta_j\|_{\mathcal{N}_\Phi}^2,\end{aligned}\quad (4)$$

subject to: $\boldsymbol{\theta}(\mathbf{x}) \in \Theta$ for all $\mathbf{x} \in \mathcal{X}$,

where the first term of the objective function measures the goodness-of-fit of the computer model's responses to the physical responses, the semi-norm $\|\theta_j\|_{\mathcal{N}_\Phi}$ measures the roughness of θ_j , and $\lambda > 0$ is a regularization/penalty parameter that balances the data fit with function smoothness. It is necessary that the solution $\hat{\boldsymbol{\theta}}(\cdot) = (\hat{\theta}_1(\cdot), \dots, \hat{\theta}_q(\cdot))$ must satisfy $\|\hat{\theta}_j\|_{\mathcal{N}_\Phi} < \infty$ so that its components are in the native space generated by Φ . We call our proposed approach the nonparametric functional calibration, since we do not assume a parametric form for the calibration function $\boldsymbol{\theta}(\cdot)$. The prediction of the outcome at a new location \mathbf{x} is obtained by the plug-in estimator $\mathbf{y}^s(\mathbf{x}, \hat{\boldsymbol{\theta}}(\mathbf{x}))$.

Sofar we have assumed the computer model is cheap, that is, $\mathbf{y}^s(\mathbf{x}, \boldsymbol{\theta})$ can be evaluated at any arbitrary point $(\mathbf{x}, \boldsymbol{\theta}) \in \mathcal{X} \times \Theta$ with almost no cost. However, in practice there is always computational cost associated with running a computer code. When the computational cost cannot be ignored when comparing with the physical experiments, we say that the computer model is expensive. To deal with the case of an expensive computer model, we can evaluate the computer model only at a set of computer design points $\mathcal{G} = \{(\mathbf{x}_1, \boldsymbol{\theta}_1), \dots, (\mathbf{x}_m, \boldsymbol{\theta}_m)\}$ and then build an emulator, or a surrogate model, based on the evaluation of the computer model at these design points (Santner, Williams, and Notz 2003). Denote such an emulator as $\hat{\mathbf{y}}_m^s(\mathbf{x}, \boldsymbol{\theta})$ which is

obtained using $\{(\mathbf{x}_j, \boldsymbol{\theta}_j, \mathbf{y}^s(\mathbf{x}_j, \boldsymbol{\theta}_j)), j = 1, \dots, m\}$. Simply replacing the computer model $\mathbf{y}^s(\mathbf{x}, \boldsymbol{\theta})$ in Equation (4) by the emulator, we obtain

$$\hat{\boldsymbol{\theta}}_{(m)}(\cdot) := \operatorname{argmin}_{\boldsymbol{\theta}(\cdot) = (\theta_1(\cdot), \dots, \theta_q(\cdot))^T} \frac{1}{n} \sum_{i=1}^n \left| \mathbf{y}_i^p - \hat{\mathbf{y}}_m^s(\mathbf{x}_i, \boldsymbol{\theta}(\mathbf{x}_i)) \right|^2 + \lambda \sum_{j=1}^q \|\theta_j\|_{\mathcal{N}_\Phi}^2, \quad (5)$$

subject to: $\boldsymbol{\theta}(\mathbf{x}) \in \Theta$ for all $\mathbf{x} \in \mathcal{X}$.

We could select the design points \mathcal{G} such that $m = n$, and for any $\mathbf{x}_i \in \mathcal{D}$ there exists $\boldsymbol{\theta}_i$ such that $(\mathbf{x}_i, \boldsymbol{\theta}_i) \in \mathcal{G}$, that is, we build the emulator based on the same design points augmented by the $\boldsymbol{\theta}$'s. However, in most applications, even expensive computer codes are much cheaper than their associated physical experiments and thus it is often reasonable to choose $m \gg n$.

Remark. The criterion functions in the optimization problems (4) and (5) can be compared with the logarithm of the posterior density presented on pages 727–728 of Brown and Atamturktur (2018). While the two terms in Equations (4) and (5) also appeared in the Bayesian formulation, the log posterior density has a few extra terms that are the result of prior specification.

3. Theoretical Properties

In this section, we develop some asymptotic properties for the proposed nonparametric functional calibration method. Here “asymptotic” means that the sample size of the physical observations tends to infinity and the error of the emulator tends to zero. In Section 3.1, we study the prediction consistency when the optimal calibration function is not required to be unique. In Section 3.2, we show the consistency of estimating the optimal calibration function when it is uniquely defined. The rates of convergence are also discussed for both scenarios.

For the rest of this section, we fix a conditionally positive-definite kernel function Φ defined on $\mathcal{X} \times \mathcal{X}$. Let $\mathcal{N}(\mathcal{X})$ be the *native space* generated by the kernel Φ with its native semi-norm $\|\cdot\|_{\mathcal{N}}$ (we drop the subscript Φ in \mathcal{N}_Φ to simplify the notation). Recall that Θ , the domain of $\boldsymbol{\theta}$, is a subset of \mathbf{R}^q . Denote $\mathcal{N}^\Theta := \{f = (f_1, \dots, f_q) : f_j \in \mathcal{N}(\mathcal{X}), j = 1, \dots, q, f(x) \in \Theta \text{ for all } x \in \mathcal{X}\}$. The functional calibration problems (4) or (5) can be cast into a unified form as

$$\hat{\boldsymbol{\theta}}_n := \operatorname{argmin}_{\boldsymbol{\theta} \in \mathcal{N}^\Theta} \frac{1}{n} \sum_{i=1}^n \left\| \mathbf{y}_i^p - \hat{\mathbf{y}}_n^s(\mathbf{x}_i, \boldsymbol{\theta}(\mathbf{x}_i)) \right\|^2 + \lambda_n \sum_{j=1}^q \|\theta_j\|_{\mathcal{N}_\Phi}^2, \quad (6)$$

for a sequence of smoothing parameters $\lambda_n > 0$, where $\{\hat{\mathbf{y}}_n^s\}$ is a sequence of emulators for \mathbf{y}^s with increasing accuracies as $n \rightarrow \infty$. For the case of cheap codes, letting $\hat{\mathbf{y}}_n^s(\mathbf{x}_i, \boldsymbol{\theta}(\mathbf{x}_i)) = \mathbf{y}^s(\mathbf{x}_i, \boldsymbol{\theta}(\mathbf{x}_i))$ in Equation (6) gives Equation (4); for the case of expensive codes, letting $\hat{\mathbf{y}}_n^s(\mathbf{x}_i, \boldsymbol{\theta}(\mathbf{x}_i)) = \hat{\mathbf{y}}_m^s(\mathbf{x}_i, \boldsymbol{\theta}(\mathbf{x}_i))$ in Equation (6) gives Equation (5). In the latter case, we allow m to depend on n . It is worth noting that if the solution of Equation (6) is not unique and we focus on prediction, our theory works for an arbitrary choice from possible solutions.

3.1. Prediction Consistency

According to Equation (3), the computer model response $\mathbf{y}^s(\mathbf{x}, \boldsymbol{\theta}^*(\mathbf{x}))$, equipped with the optimal calibration function $\boldsymbol{\theta}^*(\cdot)$, provides the best possible prediction of the physical system response $\boldsymbol{\xi}(\mathbf{x})$. The corresponding prediction error measured in L_2 distance is

$$\text{PE}(\boldsymbol{\theta}^*) = \int_{\mathcal{X}} |\boldsymbol{\xi}(\mathbf{x}) - \mathbf{y}^s(\mathbf{x}, \boldsymbol{\theta}^*(\mathbf{x}))|^2 d\mathbf{x}.$$

The L_2 norm of prediction error corresponding to the empirical calibration function defined in Equation (6) is

$$\text{PE}(\hat{\boldsymbol{\theta}}_n) = \int_{\mathcal{X}} |\boldsymbol{\xi}(\mathbf{x}) - \hat{\mathbf{y}}_n^s(\mathbf{x}, \hat{\boldsymbol{\theta}}_n(\mathbf{x}))|^2 d\mathbf{x}.$$

The difference between these two quantities tells us how well the empirical calibration function performs in terms of prediction. This subsection establishes a prediction consistency result, which states that $\text{PE}(\hat{\boldsymbol{\theta}}_n) - \text{PE}(\boldsymbol{\theta}^*) \rightarrow 0$ when the sample size $n \rightarrow \infty$. Our result does not require the uniqueness of $\boldsymbol{\theta}^*$. Note that the value of $\text{PE}(\boldsymbol{\theta}^*)$ is always uniquely defined and does not depend on the choice of feasible $\boldsymbol{\theta}^*$.

Before stating our results, we introduce some technical conditions. Call a r -dimensional random vector $\boldsymbol{\xi}$ sub-Gaussian, if there exists $\varsigma > 0$ such that

$$E[\exp\{\boldsymbol{\alpha}^T(\boldsymbol{\xi} - E\boldsymbol{\xi})\}] \leq \exp(\varsigma^2 \|\boldsymbol{\alpha}\|^2/2), \quad (7)$$

for all $\boldsymbol{\alpha} \in \mathbf{R}^r$. Here the constant ς is referred to as the sub-Gaussian parameter.

Condition 1. The design points \mathbf{x}_i 's are randomly drawn from the uniform distribution over \mathcal{X} , and the noise vectors \mathbf{e}_i 's are independent from a sub-Gaussian distribution with mean zero and sub-Gaussian parameter σ , for $i = 1, \dots, n$. Moreover, $\{\mathbf{x}_i\}_{i=1}^n$ and $\{\mathbf{e}_i\}_{i=1}^n$ are mutually independent.

Condition 2. The computer model is a smooth function of its inputs in the sense that $\mathbf{y}^s \in C^2(\mathcal{X} \times \Theta)$, the space of twice continuously differentiable functions defined on $\mathcal{X} \times \Theta$.

Define the covering number $N(\delta, \mathcal{S}, d)$ of the set \mathcal{S} as the smallest value of N for which there exist functions f_1, \dots, f_N , such that for each $f \in \mathcal{S}$, $d(f, f_j) \leq \delta$ for some $j \in \{1, \dots, N\}$. Let $\mathcal{N}(\mathcal{X}, \rho) := \{f \in \mathcal{N}(\mathcal{X}) : \|f\|_{\mathcal{N}} \leq \rho\}$ for $\rho > 0$.

Condition 3. The covering number of $\mathcal{N}(\mathcal{X}, \rho)$ satisfies

$$\log N(\epsilon, \mathcal{N}(\mathcal{X}, \rho), L_\infty(\mathcal{X})) \leq \left(\frac{C\rho}{\epsilon} \right)^{\frac{d}{\nu}}, \quad (8)$$

for some $\nu > d/2$ and a constant C independent of ϵ and ρ .

Condition 3 gives a constraint on the size of the reproducing kernel Hilbert space. Normally the value of ν depends highly on the smoothness of the kernel function. Consider the Matérn family of kernel functions (Stein 1999),

$$\begin{aligned} \Phi(s, t) &= \Phi_{\nu, \phi}(s, t) \\ &= \frac{1}{\Gamma(\nu) 2^{\nu-1}} (2\sqrt{\nu\phi} |s-t|)^\nu K_\nu(2\sqrt{\nu\phi} |s-t|), \end{aligned} \quad (9)$$

where K_ν is the modified Bessel function of the second kind. The smoothness of this kernel function is determined by the value of ν . If $\lfloor \nu + d/2 \rfloor > d/2$ (which holds if $\nu \geq 1$), the reproducing kernel Hilbert space generated by this kernel

function is equal to the (fractional) Sobolev space $H^{v+d/2}$ (Tuo and Wu 2015, coroll. 1). Using this equivalence and the covering number of the Sobolev space (Edmunds and Triebel 2008), we obtain, for $v \geq 1$, the L_∞ covering numbers of the balls of the reproducing kernel Hilbert space generated by Matérn kernel $\Phi_{v,\phi}$ are given by

$$\log N(\epsilon, \mathcal{N}_{\Phi_{v,\phi}}(\mathcal{X}, \rho), L_\infty(\mathcal{X})) \leq \left(\frac{C_{\mathcal{X}} \rho}{\epsilon} \right)^{\frac{d}{v+d/2}},$$

for a constant $C_{\mathcal{X}}$ depends on \mathcal{X} only. Condition 3 is clearly satisfied by this kernel with $v = v + d/2$. Native spaces generated by some other kernels may also admit covering number bound as Equation (8), for example, the thin-plate spline and polyharmonic spline kernels (Duchon 1977). The native spaces generated by such kernels are equivalent to certain Sobolev spaces; see Wendland (2005) for details.

We also remark that Condition 3 is satisfied for the Gaussian kernels as well. In fact, there is a much tighter entropy bound for the Gaussian case (Zhou 2002), which can yield an even faster rate of convergence. But to save space, we do not pursue this in the article.

In this work, we do not require a specific type of emulators. Users can choose their favorite emulators (such as GP regression or polynomial models) provided that they can well approximate the underlying computer response function. Condition 4 is an assumption regarding the approximation error.

Condition 4. The sequence of emulators is chosen to satisfy

$$\|\hat{y}_n^s - y^s\|_{C^1(\mathcal{X} \times \Theta)} = O(n^{-\frac{v}{2v+d}}), \quad (10)$$

where $\|f\|_{C^1(\Gamma)} := \max_{jk} \sup_{x \in \Gamma} |\frac{\partial f_k}{\partial x_j}(x)|$.

Condition 4 assumes that the magnitude of the emulation error is no bigger than that of the error of the nonparametric regression. In the case of cheap codes, $\hat{y}_n^s = y^s$, and thus Condition 4 automatically satisfied. In the case of expensive codes, Condition 4 is a requirement on the rate of approximation for the emulator. The approximation error of the emulator depends on how the emulator is constructed. For commonly used emulators, their rates of convergence are available in the numerical analysis literature. For instance, the error bound for the radial basis function interpolation can be found in Wendland (2005). Note that the input dimension of the emulator is $d+q$ because it has both control variables and calibration parameters. Suppose the smoothness of emulator is v' , then a typical rate of convergence for the emulator constructed by m computer outputs is $O(m^{-v'/(d+q)})$. Therefore, to ensure Condition 5, m should be at least with the order of magnitude $O(n^{\frac{v+d+q}{2v+d}})$. We believe this condition can be easily achieved by choosing the sample size for the computer experiment to be much larger than n , which is feasible because each run of the computer codes should be much less costly than the corresponding physical experiment.

Corresponding to the design points of the physical experiment, define an empirical (semi-)norm as $\|f\|_n = \{\sum_{i=1}^n f^2(x_i)/n\}^{1/2}$ for any function f on \mathcal{X} . To incorporate the multivariate response, we extend the notion of native norm to a vector-valued function. For $f = (f_1, \dots, f_q)$, define $\|f\|_{\mathcal{N}}$ as the Euclidean norm of $(\|f_1\|_{\mathcal{N}}, \dots, \|f_q\|_{\mathcal{N}})$. Similarly, we can also define the L_2 norm as well as the empirical norm for a vector-valued function.

Theorem 1. Suppose that Conditions 1–4 are fulfilled. In addition, we assume

$$\|\hat{y}_n^s(\mathbf{x}, \boldsymbol{\theta}(\mathbf{x}))\|_{\mathcal{N}} \leq C_1 \|\boldsymbol{\theta}\|_{\mathcal{N}} + C_2 \quad (11)$$

for all $\boldsymbol{\theta}(\cdot) \in \mathcal{N}^\Theta$ and some constants $C_1, C_2 > 0$. Then if $\lambda_n^{-1} = o_p(n^{1/2})$ and $\lambda_n = o_p(1)$, we have $\text{PE}(\hat{\boldsymbol{\theta}}_n) - \text{PE}(\boldsymbol{\theta}^*) = O_p(\lambda_n)$.

Condition (11) requires that $\hat{y}_n^s(\mathbf{x}, \boldsymbol{\theta}(\mathbf{x}))$ preserves the smoothness of \mathcal{N} , which is generally true if \hat{y}_n^s is smooth enough. For example, if \mathbf{x} and $\boldsymbol{\theta}$ are one-dimensional, Θ is compact, and \mathcal{N} is equivalent to the Sobolev space H^k with $k \in \mathbb{N}^+$, then according to (Danchin 2005, prop. 1.4.8), Equation (11) holds if \hat{y}_n^s is $k+1$ times continuously differentiable.

In Theorem 1, we consider the random design case where the design points follow the uniform distribution. The fixed design is also commonly used in practical situations, where the design points are chosen in a deterministic way. Our next result extends Theorem 1 to the fixed design case.

Condition 1'. The design points $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ are deterministic and satisfying

$$\|f\|_{L_2(\mathcal{X})} \lesssim \|f\|_n + n^{-\frac{v}{2v+d}} \|f\|_{\mathcal{N}}, \quad (12)$$

for all $f \in \mathcal{N}$. The noises e_i 's are independent from a sub-Gaussian distribution with mean zero and sub-Gaussian parameter σ , for $i = 1, \dots, n$.

The condition (12) is generally a mild condition. It holds for a broad class of design schemes called *quasi-uniform designs*, which covers many commonly used space-filling designs. For a design $\mathcal{D}_n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, define its fill distance as

$$h(\mathcal{D}_n) := \max_{\mathbf{x} \in \mathcal{X}} \min_{\mathbf{x}_i \in \mathcal{D}_n} \|\mathbf{x} - \mathbf{x}_i\|,$$

and separation distance as

$$q(\mathcal{D}_n) := \min_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_n} \|\mathbf{x}_i - \mathbf{x}_j\|.$$

Note that $h(\mathcal{D}_n)$ and $q(\mathcal{D}_n)$ are also commonly used criteria to measure the space-filling property of a design: the design minimizing $h(\mathcal{D}_n)$ is known as the *minimax design* and the design maximizing $q(\mathcal{D}_n)$ is known as the *maximin design*. See Johnson, Moore, and Ylvisaker (1990). A sequence of designs $\{\mathcal{D}_n\}$ is called quasi-uniform if $h(\mathcal{D}_n)/q(\mathcal{D}_n)$ is bounded. Utreras (1988) proved that Equation (12) is ensured if the design scheme is quasi-uniform.

Theorem 2. Assume that Condition 1', Conditions 2–4 and (11) are satisfied. Then if $\lambda_n^{-1} = o_p(n^{1/2})$ and $\lambda_n = o_p(1)$, we have $\text{PE}(\hat{\boldsymbol{\theta}}_n) - \text{PE}(\boldsymbol{\theta}^*) = O_p(\lambda_n)$.

3.2. Consistency of Calibration Function

Now we study the consistency of the empirical calibration function $\hat{\boldsymbol{\theta}}_n$ in estimating the optimal calibration function $\boldsymbol{\theta}^*$, provided the latter is uniquely defined. The required identifiability condition for $\boldsymbol{\theta}^*$ is specified below.

Condition 5. The optimal calibration function $\boldsymbol{\theta}^*(\mathbf{x})$ exists. There exists $\omega_0 > 0$, such that

$$|\zeta(\mathbf{x}) - y^s(\mathbf{x}, \boldsymbol{\theta})|^2 - |\zeta(\mathbf{x}) - y^s(\mathbf{x}, \boldsymbol{\theta}^*(\mathbf{x}))|^2 \geq \omega_0 \|\boldsymbol{\theta} - \boldsymbol{\theta}^*(\mathbf{x})\|^2, \quad (13)$$

for all $\boldsymbol{\theta} \in \Theta$ and $\mathbf{x} \in \mathcal{X}$.

Inequality (13) suggests that $|\zeta(\mathbf{x}) - y^s(\mathbf{x}, \cdot)|^2$ is bounded below locally by a quadratic function around the optimal calibration parameter $\theta^*(\mathbf{x})$. It also implies that $\theta^*(\cdot)$ is uniquely defined. A sufficient condition for Equation (13) is that Θ is a compact set, $y^s \in C^2(\mathcal{X} \times \Theta)$ and

$$\inf_{\mathbf{x} \in \mathcal{X}} \lambda_{\min} \left(\frac{\partial^2 l_x(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} \Big|_{\boldsymbol{\theta}=\theta^*(\mathbf{x})} \right) = \lambda_0 > 0, \quad (14)$$

where $l_x(\boldsymbol{\theta}) = |\zeta(\mathbf{x}) - y^s(\mathbf{x}, \boldsymbol{\theta})|^2$ and $\lambda_{\min}(A)$ denotes the smallest eigenvalue of the matrix A . To see this, we apply the Taylor expansion to obtain that for any $\boldsymbol{\theta} \in \Theta$,

$$l_x(\boldsymbol{\theta}) - l_x(\theta^*(\mathbf{x})) = (\boldsymbol{\theta} - \theta^*(\mathbf{x}))^T \frac{\partial^2 l_x(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}'} (\boldsymbol{\theta} - \theta^*(\mathbf{x})),$$

where $\boldsymbol{\theta}'$ lies between $\boldsymbol{\theta}$ and $\theta^*(\mathbf{x})$. Let $\mathbf{A} \geq \mathbf{B}$ mean $\mathbf{A} - \mathbf{B}$ is positive semidefinite for square matrices \mathbf{A}, \mathbf{B} . Because of the condition $y^s \in C^2(\mathcal{X} \times \Theta)$ and Equation (14), we can find $\delta > 0$ so that

$$\frac{\partial^2 l_x(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}'} \geq \frac{1}{2} \frac{\partial^2 l_x(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} \Big|_{\boldsymbol{\theta}=\theta^*(\mathbf{x})} \geq \frac{\lambda_0}{2} \mathbf{I}, \quad (15)$$

for all $|\boldsymbol{\theta} - \theta^*(\mathbf{x})| \leq \delta$ and $\mathbf{x} \in \mathcal{X}$. This implies

$$l_x(\boldsymbol{\theta}) - l_x(\theta^*(\mathbf{x})) \geq \lambda_0 |\boldsymbol{\theta} - \theta^*(\mathbf{x})|^2 / 2, \quad (16)$$

for all $|\boldsymbol{\theta} - \theta^*(\mathbf{x})| \leq \delta$. On the other hand, because θ^* is the unique optimal (minimum distance) calibration function, for $|\boldsymbol{\theta} - \theta^*(\mathbf{x})| > \delta$, there exists a constant $\gamma > 0$ so that

$$l_x(\boldsymbol{\theta}) - l_x(\theta^*(\mathbf{x})) \geq \gamma \geq \gamma |\boldsymbol{\theta} - \theta^*|^2 / D(\Theta)^2, \quad (17)$$

where $D(\Theta)$ denotes the diameter of Θ . Thus, Equation (13) is ensured by combining Equations (16) and (17). Note that Equation (14) is a mild condition analogous to a standard one in parametric statistical inference: the Fisher information matrix is non-singular (or positive definite).

The main result for the calibration consistency is **Theorem 3**. The rate of convergence here can be much faster than that in **Theorem 1**, and attains the minimax rate for nonparametric regression (Stone 1982). In **Theorem 3**, the smoothing parameter λ_n can be random but subject to certain order of magnitude conditions.

Theorem 3. Assume Conditions 1–5 are satisfied. Assume further that the sequence $\{\lambda_n\}$ is chosen to satisfy $\lambda_n \sim n^{-\frac{2v}{2v+d}}$, that is, $\lambda_n = O_p(n^{-\frac{2v}{2v+d}})$ and $\lambda_n^{-1} = O_p(n^{\frac{2v}{2v+d}})$. Then $\|\hat{\boldsymbol{\theta}}_n - \theta^*\|_{L_2(\mathcal{X})} = O_p(n^{-\frac{v}{2v+d}})$ and $\|\hat{\boldsymbol{\theta}}_n\|_{\mathcal{N}} = O_p(1)$.

Similar to **Theorem 2**, we have a fixed-design version of the calibration convergence theorem, given by **Theorem 4**.

Theorem 4. Assume that Condition 1' and Conditions 2–5 are satisfied. Assume further that the sequence $\{\lambda_n\}$ is chosen to satisfy $\lambda_n \sim n^{-\frac{2v}{2v+d}}$, $\lambda_n = O(n^{-\frac{2v}{2v+d}})$ and $\lambda_n^{-1} = O(n^{\frac{2v}{2v+d}})$. Then $\|\hat{\boldsymbol{\theta}}_n - \theta^*\|_{L_2(\mathcal{X})} = O_p(n^{-\frac{v}{2v+d}})$ and $\|\hat{\boldsymbol{\theta}}_n\|_{\mathcal{N}} = O_p(1)$.

Remark. Conditions 1 and 1' assume sub-Gaussian noises but this condition can be relaxed. By applying an adaptive truncation argument using Bernstein's inequality (van de Geer 2000), it

can be proved that **Theorems 1–4** hold for sub-exponential e_i 's, that is, $E[e^{c|e_i|}] < +\infty$ for some $c > 0$.

Remark. The method in this article is inspired by smoothing splines, a widely used method for nonparametric regression. However, this article addresses the problem of computer model calibration, which is a problem different from nonparametric regression. For readers interested in comparing results in this section with asymptotic results of smoothing splines, we refer to Gu (2013) and van der Geer (2000) for results of rates of convergence for smoothing splines.

4. Computation

This section develops an algorithm to solve the minimization problem (4) and a method for penalty parameter selection. The same algorithm is applicable to solve Equation (5). For simplicity, we drop the constraint $\boldsymbol{\theta}(\cdot) \in \Theta$ since we can monitor the steps of the iterative algorithm to ensure that the iteration steps will not lead to a point outside the feasible region. Moreover, we focus on the case of one-dimensional response (i.e., $r = 1$). Extension of the algorithm to general r is straightforward with only some notational complications. After these simplifications, the problem (4) becomes

$$\min_{\boldsymbol{\theta}(\cdot) = (\theta_1(\cdot), \dots, \theta_q(\cdot))^T} \sum_{i=1}^n \{y_i^p - y^s(\mathbf{x}_i, \boldsymbol{\theta}(\mathbf{x}_i))\}^2 + n\lambda \sum_{j=1}^q \|\theta_j\|_{\mathcal{N}_\Phi}^2. \quad (18)$$

Here y_i^p and $y^s(\mathbf{x}_i, \boldsymbol{\theta}(\mathbf{x}_i))$ are scalers.

We first reduce the optimization over the infinite-dimensional native space \mathcal{N}_Φ to an optimization over a finite-dimensional space. Let \mathcal{N}_0 be the finite-dimensional null space of $\mathcal{N}_\Phi(\mathcal{X})$, that is, $\mathcal{N}_0 = \{f : \|f\|_{\mathcal{N}_\Phi} = 0\}$. Let $\{v_1, v_2, \dots, v_k\}$ be a set of basis functions for \mathcal{N}_0 . The generalized representer theorem (e.g., Schölkopf, Herbrich, and Smola 2001) implies that a solution to Equation (4) has the basis expansion

$$\theta_j(\mathbf{x}) = v_j(\mathbf{x}) + \phi_j(\mathbf{x}) := \sum_{s=1}^k \alpha_{js} v_s(\mathbf{x}) + \sum_{i=1}^n \beta_{ji} \Phi(\mathbf{x}, \mathbf{x}_i). \quad (19)$$

for $j = 1, 2, \dots, q$. The solution is a sum of $v_j(\mathbf{x})$ in the null space and $\phi_j(\mathbf{x})$ in the reproducing kernel Hilbert space generated by the kernel function Φ . With the representation in Equation (19), the quasi-Newton algorithm (Byrd and Nocedal 1989) can be used to solve (18) and worked well in our numerical studies.

The penalty parameter λ can be selected in a similar manner as in the smoothing spline literature, by minimizing the generalized cross-validation or GCV criterion (Golub, Heath, and Wahba 1979; Craven and Wahba 1979). To develop an appropriate GCV criterion in our context, we extend the derivations in (Gu 2013, chap. 3).

We expand the computer model $y^s(\mathbf{x}, \boldsymbol{\theta}(\mathbf{x}))$ by its first-order Taylor expansion at the optimal value $\hat{\boldsymbol{\theta}}(\mathbf{x})$,

$$y^s(\mathbf{x}_i, \boldsymbol{\theta}(\mathbf{x}_i)) \approx y^s(\mathbf{x}_i, \hat{\boldsymbol{\theta}}(\mathbf{x}_i)) + \sum_{j=1}^q \frac{\partial}{\partial \theta_j} y^s(\mathbf{x}_i, \hat{\boldsymbol{\theta}}(\mathbf{x}_i)) \times (\theta_j(\mathbf{x}_i) - \hat{\theta}_j(\mathbf{x}_i)). \quad (20)$$

Plug this into the original objective function, we get

$$\begin{aligned} \min_{\theta(\cdot)} \sum_{i=1}^n & \left\{ y_i^p - y^s(x_i, \hat{\theta}(x_i)) - \sum_{j=1}^q \frac{\partial}{\partial \theta_j} y^s(x_i, \hat{\theta}(x_i)) \right. \\ & \times \left. \left(\theta_j(x_i) - \hat{\theta}_j(x_i) \right) \right\}^2 \\ & + n\lambda \sum_{j=1}^q \|\theta_j\|_{\mathcal{N}_\Phi}^2. \end{aligned}$$

Define $w_{ij} = \frac{\partial}{\partial \theta_j} y^s(x_i, \hat{\theta}(x_i))$ and $\bar{y}_i = y_i^p - y^s(x_i, \hat{\theta}(x_i)) + \sum_{j=1}^q w_{ij} \hat{\theta}_j(x_i)$. The original problem is then reduced to a penalized regression problem,

$$\min_{\theta(\cdot)} \sum_{i=1}^n \left\{ \bar{y}_i - \sum_{j=1}^q w_{ij} \theta_j(x_i) \right\}^2 + n\lambda \sum_{j=1}^q \|\theta_j\|_{\mathcal{N}_\Phi}^2. \quad (21)$$

Let $\alpha_j = (\alpha_{j1}, \dots, \alpha_{jk})^T$ and $\beta_j = (\beta_{j1}, \dots, \beta_{jn})^T$. The reproducing property of the kernel function Φ implies that $\|\theta_j\|_{\mathcal{N}_\Phi}^2 = \beta_j^T \Phi \beta_j$, where $\Phi = (\Phi(x_i, x_j))_{n \times n}$. Let $\bar{y} = (\bar{y}_1, \dots, \bar{y}_n)^T$, $\mathbf{V} = (v_s(x_i))_{n \times q}$, and the diagonal weight matrix $\mathbf{W}_j = \text{diag}(w_{1j}, \dots, w_{nj})$. The optimization problem (21) is expressed in the matrix notation as

$$\min_{\alpha_j, \beta_j} \left\| \bar{y} - \sum_{j=1}^q \mathbf{W}_j (\mathbf{V} \alpha_j + \Phi \beta_j) \right\|_2^2 + n\lambda \sum_{j=1}^q \beta_j^T \Phi \beta_j, \quad (22)$$

which in turn is equivalent to

$$\min_{\alpha_j, \beta_{jw}} \left\| \bar{y} - \sum_{j=1}^q (\mathbf{V}_{jw} \alpha_j + \Phi_{jw} \beta_{jw}) \right\|_2^2 + n\lambda \sum_{j=1}^q \beta_{jw}^T \Phi_{jw} \beta_{jw}. \quad (23)$$

where $\mathbf{V}_{jw} = \mathbf{W}_j \mathbf{V}$, $\Phi_{jw} = \mathbf{W}_j \Phi \mathbf{W}_j$, and $\beta_{jw} = \mathbf{W}_j^{-1} \beta_j$.

For the objective function in Equation (23), take first order derivative with respect to α_j and β_{jw} and set them to zero. We find that the optimal solution must satisfy

$$\mathbf{V}_{jw}^T \left\{ \bar{y} - \sum_{j=1}^q (\mathbf{V}_{jw} \alpha_j + \Phi_{jw} \beta_{jw}) \right\} = \mathbf{0}, \quad (24)$$

$$\Phi_{jw} \left\{ \bar{y} - \sum_{j=1}^q (\mathbf{V}_{jw} \alpha_j + \Phi_{jw} \beta_{jw}) \right\} - n\lambda \Phi_{jw} \beta_{jw} = \mathbf{0}. \quad (25)$$

If Φ_{jw} is singular, the above equation may have multiple solutions. It is easy to see Equations (24) and (25) are satisfied by the solution of

$$\mathbf{V}_{kw}^T \beta_{jw} = \mathbf{0}, \quad (26)$$

$$\bar{y} - \sum_{j=1}^q (\mathbf{V}_{jw} \alpha_j + \Phi_{jw} \beta_{jw}) - n\lambda \beta_{jw} = \mathbf{0}, \quad (27)$$

for any $j, k = 1, 2, \dots, q$ and the solution is unique. In fact, pre-multiply (27) by Φ_{jw} , we get Equation (25). In addition, Equation (26) for $k = j$ and Equation (27) together imply Equation (24).

These equations can be arranged into a more compact form. Let $\alpha = (\alpha_1^T, \alpha_2^T, \dots, \alpha_q^T)^T$, and $\beta_w = (\beta_{1w}^T, \beta_{2w}^T, \dots, \beta_{qw}^T)^T$. We also let \bar{y} be repeated q times, and the resulting vector is denoted as $\mathbf{Y} = (\bar{y}^T, \dots, \bar{y}^T)^T \in \mathbf{R}^{qn}$. Similarly, we define

$$\begin{aligned} \mathbf{V}_w &= \begin{pmatrix} \mathbf{V}_{1w} & \mathbf{V}_{2w} & \cdots & \mathbf{V}_{qw} \\ \mathbf{V}_{1w} & \mathbf{V}_{2w} & \cdots & \mathbf{V}_{qw} \\ \vdots & \vdots & & \vdots \\ \mathbf{V}_{1w} & \mathbf{V}_{2w} & \cdots & \mathbf{V}_{qw} \end{pmatrix}, \\ \Phi_w &= \begin{pmatrix} \Phi_{1w} & \Phi_{2w} & \cdots & \Phi_{qw} \\ \Phi_{1w} & \Phi_{2w} & \cdots & \Phi_{qw} \\ \vdots & \vdots & & \vdots \\ \Phi_{1w} & \Phi_{2w} & \cdots & \Phi_{qw} \end{pmatrix}. \end{aligned}$$

where the row blocks simply repeat the first row block q times. It follows that Equations (26) and (27) can be presented as

$$\mathbf{V}_w^T \beta_w = \mathbf{0}, \quad (28)$$

$$\mathbf{Y} - \mathbf{V}_w \alpha - (\Phi_w + n\lambda \mathbf{I}) \beta_w = \mathbf{0}, \quad (29)$$

This requires β_w to be orthogonal to the column space of \mathbf{V}_w . Suppose the full QR decomposition of \mathbf{V}_w is $\mathbf{V}_w = (\mathbf{F}_1 \mathbf{F}_2)(\mathbf{R}^T \mathbf{0}^T)^T$, then we have $\mathbf{F}_1^T \beta_w = 0$ and $\beta_w = \mathbf{F}_2 \mathbf{F}_2^T \beta_w$. Pre-multiply (29) by \mathbf{F}_2^T and \mathbf{F}_1^T separately, the solution to Equations (28) and (29) can be found as

$$\hat{\beta}_w = \mathbf{F}_2 (\mathbf{F}_2^T \Phi_w \mathbf{F}_2 + n\lambda \mathbf{I})^{-1} \mathbf{F}_2^T \mathbf{Y}, \quad (30)$$

$$\hat{\alpha} = \mathbf{R}^{-1} (\mathbf{F}_1^T \mathbf{Y} - \mathbf{F}_1^T \Phi_w \hat{\beta}_w). \quad (31)$$

Plug the above solution into Equation (29) and with some calculation, we obtain that the estimated \mathbf{Y} is $\hat{\mathbf{Y}} = \mathbf{V}_w \hat{\alpha} + \Phi_w \hat{\beta}_w = \mathbf{A}(\lambda) \mathbf{Y}$, where $\mathbf{A}(\lambda) = \mathbf{I} - n\lambda \mathbf{F}_2 (\mathbf{F}_2^T \Phi_w \mathbf{F}_2 + n\lambda \mathbf{I})^{-1} \mathbf{F}_2^T$.

The smoothing matrix $\mathbf{A}(\lambda)$ transforms \mathbf{Y} to $\hat{\mathbf{Y}} = \mathbf{A}(\lambda) \mathbf{Y}$. This connection suggests the following generalized cross-validation (GCV) criterion

$$\text{GCV}(\lambda) = \frac{(qn)^{-1} \mathbf{Y}^T (\mathbf{I} - \mathbf{A}(\lambda))^2 \mathbf{Y}}{\{(qn)^{-1} \text{tr}(\mathbf{I} - \mathbf{A}(\lambda))\}^2}. \quad (32)$$

The term q appears in the above equation because the whole set of observations is repeated q times in our derivation. This GCV is employed to select the penalty parameter λ .

5. Uncertainty Quantification

The uncertainty of estimation of the calibration function and prediction comes from two sources: the physical experiments and the computer experiments. We develop an uncertainty quantification method that considers only the former source of uncertainty, which is usually the dominant source of uncertainty. This method is adequate for use in the case of cheap codes, and may under-estimate the uncertainty in the case of expensive codes. As in the previous section, our presentation focuses on the case of one-dimensional responses (i.e., $r = 1$).

Adopting a strategy typically used in the smoothing spline regression literature (e.g., Wahba 1990; Gu 2013), we develop a confidence interval for $\hat{\theta}(\mathbf{x})$ at an arbitrary \mathbf{x} via considering the penalized regression (21) in the Bayesian framework. Let σ_e^2

denote the variance of the measurement noise e_i in data model (1). The objective function of Equation (21) is proportional to

$$\sum_{i=1}^n -\frac{1}{2\sigma_e^2} \left\{ \bar{y}_i - \sum_{j=1}^q w_{ij}\theta_j(x_i) \right\}^2 - \frac{n\lambda}{2\sigma_e^2} \sum_{j=1}^q \|\theta_j\|_{\mathcal{N}_\Phi}^2.$$

Taking exponential of the above, we arrive at

$$\begin{aligned} & \prod_{i=1}^n \exp \left(-\frac{1}{2\sigma_e^2} \left\{ \bar{y}_i - \sum_{j=1}^q w_{ij}\theta_j(x_i) \right\}^2 \right) \quad (33) \\ & \times \prod_{j=1}^q \exp \left(-\frac{n\lambda}{2\sigma_e^2} \|\theta_j\|_{\mathcal{N}_\Phi}^2 \right). \end{aligned}$$

First, recall that $w_{ij} = \frac{\partial}{\partial \theta_j} y^s(x_i, \hat{\theta}(x_i))$ and $\bar{y}_i = y_i^p - y^s(x_i, \hat{\theta}(x_i)) + \sum_{j=1}^q w_{ij}\hat{\theta}_j(x_i)$. When the estimated $\hat{\theta}$ is close to the population optimal θ^* , the distribution of \bar{y}_i is close to that of

$$y_i^p - y^s(x_i, \theta^*(x_i)) + \sum_{j=1}^q w_{ij}\theta_j^*(x_i) = \sum_{j=1}^q w_{ij}\theta_j^*(x_i) + \epsilon_i. \quad (34)$$

In other words, the term \bar{y}_i approximately follows a normal distribution with mean $\sum_{j=1}^q w_{ij}\theta_j^*(x_i)$ and variance σ^2 , which is the first term in Equation (33).

Second, the penalty on $\theta_j(\cdot)$ is equivalent to imposing a prior on the native space \mathcal{N}_Φ corresponding to kernel Φ . Consider the decomposition (19), for the function $\phi_j(\cdot)$ in the reproducing kernel Hilbert space, one can think it follows a GP prior $\phi_j(\cdot) \sim \mathcal{GP}(0, \frac{\sigma_e^2}{n\lambda} \Phi(\cdot, \cdot))$. Lastly, we can impose an additional prior on the coefficients $\alpha_{js} \sim N(0, \rho \sigma_e^2 / (n\lambda))$. The constant ρ takes a large value, resulting in an approximately non-informative prior on the null space. Combining the above leads to the joint distribution (33) when ρ approaches infinity.

At an arbitrary location \mathbf{x} , define $\mathbf{v} = (v_1(\mathbf{x}), \dots, v_s(\mathbf{x}))^T$ and $\boldsymbol{\phi} = (\Phi(\mathbf{x}, \mathbf{x}_1), \dots, \Phi(\mathbf{x}, \mathbf{x}_n))^T$. The joint distribution of $\theta_j(\mathbf{x})$ and \bar{y} is a normal distribution

$$\mathcal{N} \left(\mathbf{0}, \frac{\sigma_e^2}{n\lambda} \begin{pmatrix} \sigma_{11} & \boldsymbol{\Sigma}_{21}^T \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix} \right),$$

where $\sigma_{11} = \Phi(\mathbf{x}, \mathbf{x}) + \rho \mathbf{v}^T \mathbf{v}$, $\boldsymbol{\Sigma}_{21} = \mathbf{W}_j \boldsymbol{\phi} + \rho \mathbf{V}_{jw} \mathbf{v}$, and $\boldsymbol{\Sigma}_{22} = \sum_{j=1}^q (\Phi_{jw} + \rho \mathbf{V}_{jw} \mathbf{V}_{jw}^T) + n\lambda \mathbf{I}$. It follows that the conditional variance of $\theta_j(\mathbf{x})$ given \bar{y} is

$$\hat{\sigma}_{\theta_j}^2(\mathbf{x}) = \frac{\sigma_e^2}{n\lambda} (\sigma_{11} - \boldsymbol{\Sigma}_{21}^T \boldsymbol{\Sigma}_{22}^{-1} \boldsymbol{\Sigma}_{21}). \quad (35)$$

To make use of this distribution result, we need an estimate of σ_e^2 . Based on the converted regression problem (23), we adopt the following estimator (Gu 2013, eq. 3.26)

$$\hat{\sigma}_e^2 = \frac{\mathbf{Y}_w^T (\mathbf{I} - \mathbf{A}(\lambda))^2 \mathbf{Y}_w}{\text{tr}(\mathbf{I} - \mathbf{A}(\lambda))}, \quad (36)$$

and λ is chosen by minimizing the GCV criterion (32).

Putting the above together, the $(1 - \alpha) \times 100\%$ confidence interval for $\theta_j(\mathbf{x})$ is

$$(\hat{\theta}_j(\mathbf{x}) - z_{\alpha/2} \hat{\sigma}_{\theta_j}(\mathbf{x}), \hat{\theta}_j(\mathbf{x}) + z_{\alpha/2} \hat{\sigma}_{\theta_j}(\mathbf{x})), \quad (37)$$

where z_α is the upper α quantile of the standard normal distribution and $\hat{\sigma}_{\theta_j}(\mathbf{x})$ is calculated using the expression (35) with σ_e^2 estimated by Equation (36) and λ selected by the GCV. The confidence intervals for functional calibration parameters given in (37) only make sense when the optimal calibration functions are identifiable.

While the conditional variance of $\theta(\mathbf{x})$ given \bar{y} can be computed as Equation (37), the conditional variance of $y^s(\mathbf{x}, \theta(\mathbf{x}))$ given \bar{y} , denoted as $\sigma_{y^s}^2(\mathbf{x})$, is easily obtained by the delta method. In particular, when the gradient is non-zero, that is, $\frac{\partial}{\partial \theta} y^s(\mathbf{x}, \hat{\theta}(\mathbf{x})) \neq \mathbf{0}$, we can compute

$$\sigma_{y^s}^2(\mathbf{x}) = \frac{\sigma_e^2}{n\lambda} (\Lambda_{11} - \boldsymbol{\Lambda}_{21}^T \boldsymbol{\Lambda}_{22}^{-1} \boldsymbol{\Lambda}_{21}), \quad (38)$$

where

$$\begin{aligned} \Lambda_{11} &= \sum_{j=1}^q w_j^2 (\Phi(\mathbf{x}, \mathbf{x}) + \rho \mathbf{v}^T \mathbf{v}), \\ \boldsymbol{\Lambda}_{21} &= \sum_{j=1}^q w_j (\mathbf{W}_j \boldsymbol{\phi} + \rho \mathbf{V}_{jw} \mathbf{v}), \\ \boldsymbol{\Lambda}_{22} &= \sum_{j=1}^q (\Phi_{jw} + \rho \mathbf{V}_{jw} \mathbf{V}_{jw}^T) + n\lambda \mathbf{I}, \end{aligned}$$

and $w_j = \frac{\partial}{\partial \theta_j} y^s(\mathbf{x}, \hat{\theta}(\mathbf{x}))$. This expression can be equivalently obtained from the covariance matrix of $y^s(\mathbf{x}, \theta(\mathbf{x}))$ and \bar{y} , which is computed with the aid of (20) to be

$$\frac{\sigma_e^2}{n\lambda} \boldsymbol{\Lambda} = \frac{\sigma_e^2}{n\lambda} \begin{pmatrix} \Lambda_{11} & \boldsymbol{\Lambda}_{21}^T \\ \boldsymbol{\Lambda}_{21} & \boldsymbol{\Lambda}_{22} \end{pmatrix}.$$

The $(1 - \alpha) \times 100\%$ confidence interval for the computer model response $y^s(\mathbf{x}, \theta^*(\mathbf{x}))$ at location \mathbf{x} is

$$(y^s(\mathbf{x}, \hat{\theta}(\mathbf{x})) - z_{\alpha/2} \hat{\sigma}_{y^s}(\mathbf{x}), y^s(\mathbf{x}, \hat{\theta}(\mathbf{x})) + z_{\alpha/2} \hat{\sigma}_{y^s}(\mathbf{x})), \quad (39)$$

where $\hat{\sigma}_{y^s}(\mathbf{x})$ is calculated using Equation (38) with σ_e^2 estimated by using Equation (36). This can be used as a confidence interval for the physical response $\zeta(\mathbf{x})$. The confidence interval of the physical response given in (39) is meaningful even when $\theta^*(\mathbf{x})$ is unidentified, since the optimal prediction function $y^s(\mathbf{x}, \theta^*(\mathbf{x}))$ is uniquely defined.

We expect that the confidence intervals in Equations (37) and (39) have the across-the-function coverage property (Wahba 1983; Nychka 1988). Rigorous asymptotic justification is left for future research. In the next section, we will empirically illustrate the performance of these intervals in a simulation study.

6. Simulation Study

In this section, we compare our proposed method, the non-parametric functional calibration, with the constant calibration (abbreviated as Const; Tuo and Wu 2016), the parametric functional calibration (Pourhabib et al. 2015), the Bayesian method of Brown and Atamturktur (2018). To set up the stage for comparison, we consider two parametric calibration models in Section 6.1 and two unidentified calibration models in Section 6.2. For parametric functional calibration, we consider

two parametric models. The first is the parametric exponential (Param-Exp) model, $\theta(x) = \gamma_0 \exp(\gamma_1 x)$. The second is a quadratic model (Param-Quad) of the form $\theta(x) = \gamma_0 + \gamma_1 x + \gamma_2 x^2$. For our proposed method, the native space is chosen as the Sobolev space of order 2. It has kernel $k_2(x, y) = \frac{1}{(2!)^2} B_2(x)B_2(y) - \frac{1}{4!} B_4(|x-y|)$, where $B_m(\cdot)$ is the m th Bernoulli polynomial (see, e.g., Gu 2013, p. 39). This kernel is commonly used in the smoothing spline literature and the solution in this space is a cubic spline function. In the following, our nonparametric functional calibration method using this kernel is denoted as RKHS-Cubic.

In the comparative study of Section 6.2 for the cases of unidentified calibration models, we also include local approximate GP regression method (abbreviated as laGP; Gramacy et al. 2015; Gramacy 2016), which runs a GP regression on the residuals from a constant calibration model. The laGP method is a mis-specified model in the context of Section 6.1 and thus not considered in that section where the estimation error of the calibration parameter is evaluated.

In our study, we consider both cases of cheap code and expensive code. This allows us to evaluate the impact of using the emulator on each method in the case of expensive code. In the cheap code cases, the exact computer model $y^s(x, \theta)$ are used for calibration. In the expensive code cases, the computer model is evaluated on a grid of size 14×15 on the domain of interest and a GP emulator with the squared exponential kernel (e.g., Rasmussen and Williams 2006, p. 83) is trained on the computer generated data. The output of the emulator serves as a surrogate computer model for calibration.

6.1. Two Parametric Models

For each of the two settings, one of the above two parametric calibration models (Param-Exp or Param-Quad) exactly matches the physical model for a given parameter. This allows us to compare the performance of the parametric calibration

model when the model is correctly specified and also when the model is misspecified.

1. The first setting. The physical response is $y^p(x) = \exp(x/10) \cos(x) + \sigma e$ for $x \in [\pi, 3\pi]$, where $\sigma = 0.1$ and $e \sim N(0, 1)$. The computer model is $y^s(x, \theta) = 0.5 \exp(x/10) \cos(x) \frac{\exp(x/5)}{\theta}$ with the calibration parameter θ . The optimal calibration function is $\theta^*(x) = 0.5 \exp(x/5)$. The emulator is trained on the domain $[\pi, 3\pi] \times [\pi/5, 6\pi/5]$.
2. The second setting. The physical response is $y^p(x) = \cos(2x) \sin(x/2) + \sigma e$ for $x \in [0.5\pi, \pi]$, where $\sigma = 0.1$ and $e \sim N(0, 1)$. The computer model is

$$y^s(x, \theta) = \cos(2x) \sin(x/2) \exp\left(\frac{3\theta}{0.5(x-2)^2 + 0.5} - 3\right)$$

with the calibration parameter θ . The optimal calibration function is $\theta^*(x) = 0.5(x-2)^2 + 0.5$. The emulator is trained on the domain $[0.5\pi, \pi] \times [\pi/9, \pi/2]$.

For both the cheap and expensive code cases in these two settings, the simulation is repeated 100 times. In each replication, $n = 50$ sample points are generated from the physical response model with the design points x_i 's uniformly generated on the domain.

The following metrics are employed to compare different methods. The accuracy of the estimate $\hat{\theta}(x)$ is measured by the L_2 -loss, $\{\int_{\mathcal{X}} (\theta^*(x) - \hat{\theta}(x))^2 dx\}^{1/2}$. For each method, confidence intervals are constructed for three nominal levels of coverage probability, 90%, 95% and 99%. Suppose the upper and lower bound of the confidence interval are $U(x)$ and $L(x)$. Its average width across the domain \mathcal{X} is computed as $\int_{\mathcal{X}} \{U_{0.05}(x) - L_{0.05}(x)\} dx$. The average coverage rate (CR) across \mathcal{X} is measured as $\int_{\mathcal{X}} I(U_{\alpha}(x) > \theta^*(x) > L_{\alpha}(x)) dx / |\mathcal{X}|$. The integrals are approximated by the Riemann sum with 200 equally spaced points on the domain.

The results are summarized in Tables 1 and 2. In term of estimation accuracy, the correctly specified parametric calibration model gives the best result, RKHS-Cubic and the Bayesian

Table 1. Simulation 1 (Param-Exp model).

Code	Method	L_2 -loss	90% Width	CR	95% Width	CR	99% Width	CR
CC	Const	2.222 (0.022)	2.376 (0.027)	0.123 (0.001)	2.831 (0.032)	0.146 (0.002)	3.721 (0.042)	0.193 (0.002)
	Param-Exp	0.063 (0.003)	0.527 (0.006)	0.883 (0.021)	0.628 (0.007)	0.946 (0.014)	0.825 (0.010)	0.989 (0.007)
	Param-Quad	0.086 (0.003)	0.620 (0.008)	0.822 (0.020)	0.739 (0.009)	0.891 (0.016)	0.971 (0.012)	0.966 (0.009)
	RKHS-Cubic	0.131 (0.004)	1.074 (0.021)	0.895 (0.012)	1.280 (0.025)	0.937 (0.010)	1.682 (0.033)	0.979 (0.005)
	Baysian	0.157 (0.005)	1.696 (0.019)	0.960 (0.006)	2.030 (0.022)	0.979 (0.004)	2.693 (0.029)	0.994 (0.001)
EC	Const	2.288 (0.024)	2.550 (0.027)	0.129 (0.002)	3.038 (0.033)	0.155 (0.002)	3.993 (0.043)	0.204 (0.002)
	Param-Exp	0.072 (0.005)	0.535 (0.007)	0.830 (0.024)	0.637 (0.008)	0.903 (0.019)	0.837 (0.010)	0.965 (0.012)
	Param-Quad	0.095 (0.005)	0.643 (0.008)	0.827 (0.021)	0.766 (0.010)	0.891 (0.017)	1.007 (0.013)	0.955 (0.011)
	RKHS-Cubic	0.164 (0.007)	1.296 (0.030)	0.890 (0.011)	1.544 (0.036)	0.935 (0.009)	2.029 (0.047)	0.981 (0.005)
	Baysian	0.165 (0.006)	1.762 (0.018)	0.948 (0.007)	2.108 (0.022)	0.972 (0.005)	2.792 (0.030)	0.996 (0.002)

NOTES: Comparison of methods with the cheap code (CC) and expensive code (EC). The mean (and SE) of the L_2 -loss and of the width and average coverage rate (CR) for the 90%, 95%, and 99% confidence intervals.

Table 2. Simulation 2 (Param-Quad model).

Code	Method	L_2 -loss	90% Width	CR	95% Width	CR	99% Width	CR
	Const	0.278 (0.001)	0.075 (0.001)	0.202 (0.003)	0.089 (0.001)	0.245 (0.004)	0.117 (0.001)	0.341 (0.007)
	Param-Exp	0.094 (0.000)	0.080 (0.001)	0.192 (0.003)	0.096 (0.001)	0.230 (0.003)	0.126 (0.002)	0.306 (0.005)
	Param-Quad	0.011 (0.000)	0.048 (0.001)	0.882 (0.018)	0.057 (0.001)	0.944 (0.012)	0.075 (0.001)	0.988 (0.004)
	RKHS-Cubic	0.019 (0.001)	0.076 (0.001)	0.898 (0.010)	0.091 (0.001)	0.945 (0.007)	0.119 (0.002)	0.985 (0.003)
	Bayesian	0.025 (0.001)	0.116 (0.001)	0.948 (0.004)	0.138 (0.001)	0.963 (0.003)	0.182 (0.001)	0.978 (0.002)
	Const	0.279 (0.001)	0.072 (0.001)	0.196 (0.003)	0.086 (0.001)	0.240 (0.005)	0.113 (0.001)	0.328 (0.006)
	Param-Exp	0.095 (0.000)	0.079 (0.001)	0.186 (0.002)	0.094 (0.001)	0.223 (0.003)	0.124 (0.001)	0.297 (0.004)
	Param-Quad	0.011 (0.001)	0.048 (0.001)	0.889 (0.018)	0.058 (0.001)	0.939 (0.014)	0.076 (0.001)	0.982 (0.008)
	RKHS-Cubic	0.019 (0.001)	0.074 (0.001)	0.891 (0.011)	0.088 (0.001)	0.940 (0.007)	0.115 (0.002)	0.986 (0.003)
	Bayesian	0.026 (0.001)	0.115 (0.001)	0.947 (0.004)	0.137 (0.001)	0.959 (0.003)	0.181 (0.001)	0.976 (0.002)

NOTES: Comparison of methods with the cheap code (CC) and expensive code (EC). The mean (and SE) of the L_2 -loss and of the width and average coverage rate (CR) for the level 90%, 95% and 99% confidence intervals.

methods have comparable performance, whereas the Const method performs the worst with no surprise. The results for uncertainty quantification are more interesting. In term of the coverage rate close to nominal level, RKHS-Cubic is competitive to the correctly specified parametric calibration in the case of cheap code, and outperforms in the case of expensive code. The confidence intervals produced by misspecified parametric calibration can have very low coverage. Comparing two nonparametric functional calibration methods, the actual confidence interval coverage rate for RKHS-Cubic is closer to the nominal coverage rate, whereas the Bayesian method is conservative and produces much wider intervals with coverage rate higher than the nominal level.

6.2. Two Unidentified Models

We consider two settings that the optimal calibration function is not uniquely defined. Since parameter/function estimation is meaningless in this situation, we focus on prediction performance of the competing methods.

- The third setting. The physical response is $y^p(x) = 1+x^3+\sigma e$ for $x \in [1, 2]$, where $\sigma = 0.2$ and $e \sim N(0, 1)$. The computer model is $y^s(x, \theta) = y^s(x, \theta) = \theta_1 x + \theta_2 x^2$ with two calibration parameters θ_1 and θ_2 . The functional calibration problem is not identifiable. For example, one possible solution is $\theta_1(x) = 1/x$ and $\theta_2(x) = x$, another possible solution is $\theta_1(x) = x^2$ and $\theta_2(x) = 1/x^2$. Plugging either solution into the computer model gives a model that matches exactly the physical response function.
- The fourth setting. The physical response is $y^p(x) = x^3 + \sigma e$ for $x \in [1, 2]$, where $\sigma = 0.2$ and $e \sim N(0, 1)$. The computer model is $y^s(x, \theta) = \theta_1 x^{\theta_2}$ with two calibration parameters θ_1 and θ_2 . The functional calibration problem is non-identifiable because for any $\alpha \in \mathbb{R}$, the pair $\theta_1(x) = x^\alpha$ and $\theta_2(x) = 3 - \alpha$ constitutes a possible solution.

For both the cheap and expensive code cases in these two settings, the simulation is repeated 100 times. In each replication,

$n = 50$ sample points are generated from on the physical response model with the design points x_i 's uniformly generated on the domain.

The target of prediction is the physical response function $y^p(x)$. To measure the qualify of the calibrated computer model $y^s(x, \hat{\theta}(x))$ in predicting $y^p(x)$, we use the L_2 -loss for prediction, that is, $\{\int_{\mathcal{X}} [y^p(x) - y^s(x, \hat{\theta}(x))]^2 dx\}^{1/2}$. The average width and average coverage rate for predicting the physical response function can be defined similarly as that for estimating the calibration function, used in Tables 1 and 2. Again, the integrals are approximated by the Riemann sum with 200 equally spaced points on the domain.

The results for several competing methods are summarized in Tables 3 and 4. The three methods, Const, Param-Exp, Param-Exp, do not suffer from the identifiability problem since these models impose rigid forms on the calibration function. Our algorithm for our RKHS-Cubic method converges to different local solutions with random initial starting values, but the resulting prediction is rather stable. For the third setting, our RKHS-Cubic method gives the most accurate prediction in terms of the L_2 -loss for prediction, while the Bayesian method completely fails in prediction in the expensive code case, giving a huge L_2 -loss. For the fourth setting, our RKHS-Cubic method outperforms other competing methods in prediction except the Const method and the laGP method. That the Const model performs the best in this setting is not surprising, because this setup is in favor of the Const model: The constant calibration parameters $\theta_1 = 1$ and $\theta_2 = 3$ give the optimal solution, and the Const method is constrained to only search over a 2-dimensional parameter set rather than an infinite-dimensional function space for the optimal solution. The laGP method is built on the Const model and thus has its advantage. It is interesting to note that the performance of the laGP method in terms of L_2 -loss deteriorates significantly from the cheap code to the expensive code case.

The results for uncertainty quantification in these unidentified settings are more interesting. Our RKHS-Cubic method

Table 3. Simulation 3. Comparison of methods with the cheap code (CC) and expensive code (EC). The mean (and SE) of the L_2 -loss and of the width and average coverage rate (CR) for the level 90%, 95%, and 99% confidence intervals.

Code	Method	L_2 -loss	90% Width	CR	95% Width	CR	99% Width	CR
CC	Const	0.151 (0.001)	0.158 (0.002)	0.298 (0.003)	0.188 (0.002)	0.358 (0.004)	0.247 (0.002)	0.477 (0.006)
	Param-Exp	0.060 (1.016)	0.733 (0.384)	0.852 (0.018)	0.874 (0.457)	0.911 (0.014)	1.148 (0.601)	0.967 (0.009)
	Param-Quad	0.052 (0.002)	0.181 (0.002)	0.899 (0.015)	0.216 (0.003)	0.944 (0.012)	0.284 (0.003)	0.985 (0.006)
	RKHS-Cubic	0.044 (0.002)	0.158 (0.002)	0.908 (0.015)	0.188 (0.002)	0.953 (0.011)	0.247 (0.003)	0.986 (0.006)
	Bayesian	0.067 (0.002)	0.885 (0.001)	1.000 (0.000)	1.054 (0.001)	1.000 (0.000)	1.381 (0.001)	1.000 (0.000)
	laGP	0.057 (0.003)	0.667 (0.007)	1.000 (0.000)	0.795 (0.009)	1.000 (0.000)	1.044 (0.012)	1.000 (0.000)
	Const	0.216 (0.010)	0.125 (0.003)	0.184 (0.006)	0.149 (0.004)	0.223 (0.008)	0.196 (0.005)	0.300 (0.011)
EC	Param-Exp	0.481 (0.100)	0.302 (0.040)	0.605 (0.025)	0.360 (0.048)	0.670 (0.025)	0.473 (0.063)	0.775 (0.023)
	Param-Quad	0.220 (0.025)	0.283 (0.017)	0.735 (0.021)	0.337 (0.021)	0.802 (0.019)	0.443 (0.027)	0.888 (0.014)
	RKHS-Cubic	0.114 (0.015)	0.311 (0.028)	0.879 (0.017)	0.370 (0.033)	0.915 (0.015)	0.486 (0.043)	0.954 (0.011)
	Bayesian	2.326 (0.054)	9.000 (0.289)	0.891 (0.016)	9.982 (0.360)	0.916 (0.015)	11.206 (0.470)	0.934 (0.014)
	laGP	0.228 (0.024)	0.684 (0.008)	0.967 (0.003)	0.815 (0.009)	0.981 (0.002)	1.071 (0.012)	0.991 (0.001)

Table 4. Original Simulation 4. Comparison of methods with the cheap code (CC) and expensive code (EC). The mean (and SE) of the L_2 -loss and of the width and average coverage rate (CR) for the level 90%, 95% and 99% confidence intervals.

Code	Method	L_2 -loss	90% Width	CR	95% Width	CR	99% Width	CR
CC	Const	0.035 (0.002)	0.125 (0.001)	0.883 (0.023)	0.149 (0.002)	0.928 (0.018)	0.196 (0.002)	0.982 (0.008)
	Param-Exp	0.060 (0.003)	0.180 (0.004)	0.859 (0.017)	0.215 (0.005)	0.914 (0.013)	0.282 (0.006)	0.975 (0.007)
	Param-Quad	0.069 (0.002)	0.220 (0.004)	0.889 (0.013)	0.262 (0.005)	0.940 (0.009)	0.345 (0.007)	0.988 (0.004)
	RKHS-Cubic	0.058 (0.002)	0.181 (0.002)	0.879 (0.017)	0.216 (0.002)	0.928 (0.013)	0.284 (0.003)	0.977 (0.008)
	Bayesian	0.076 (0.003)	0.820 (0.008)	0.998 (0.001)	0.978 (0.009)	0.999 (0.001)	1.286 (0.013)	1.000 (0.000)
	laGP	0.037 (0.002)	0.643 (0.007)	1.000 (0.000)	0.766 (0.008)	1.000 (0.000)	1.007 (0.011)	1.000 (0.000)
	Const	0.035 (0.002)	0.129 (0.001)	0.898 (0.021)	0.154 (0.002)	0.955 (0.014)	0.202 (0.002)	0.992 (0.004)
EC	Param-Exp	0.082 (0.004)	0.175 (0.004)	0.740 (0.020)	0.208 (0.004)	0.805 (0.017)	0.274 (0.006)	0.886 (0.012)
	Param-Quad	0.066 (0.002)	0.245 (0.014)	0.893 (0.012)	0.291 (0.016)	0.937 (0.009)	0.383 (0.021)	0.984 (0.004)
	RKHS-Cubic	0.057 (0.002)	0.180 (0.002)	0.877 (0.016)	0.214 (0.003)	0.937 (0.012)	0.281 (0.004)	0.978 (0.006)
	Bayesian	0.061 (0.002)	0.944 (0.002)	0.986 (0.003)	1.125 (0.002)	0.994 (0.002)	1.477 (0.003)	0.999 (0.000)
	laGP	0.055 (0.002)	0.645 (0.007)	1.000 (0.000)	0.768 (0.009)	1.000 (0.000)	1.010 (0.011)	1.000 (0.000)

gives comparatively shorter confidence intervals with actual coverage rate close to the nominal coverage rate (with a slight under-coverage). Even in the cases when the existing Bayesian method (Brown and Atamturktur 2018) gives comparable prediction errors, its confidence intervals (for prediction) are substantially wider than those produced by our proposed method. Inspecting the posterior distributions of the functional calibration parameters produced by the MCMC indicates that the MCMC samples diffuse over many local modes; see Figure 1. This could be one of the main reasons why the Bayesian method produces substantially wider confidence intervals. Deeper understanding of the behavior of the Bayesian

method in unidentified settings needs further research. The confidence intervals provided by the laGP method are shorter than the corresponding intervals of the Bayesian methods but are still much wider than those provided by our RKHS-Cubic method.

7. Real Data: Young's Modulus Prediction in Buckypaper Fabrication

Pourhabib et al. (2015) proposed parametric functional calibration as a method for modulus prediction in buckypaper fabrication. Below we apply the proposed nonparametric functional

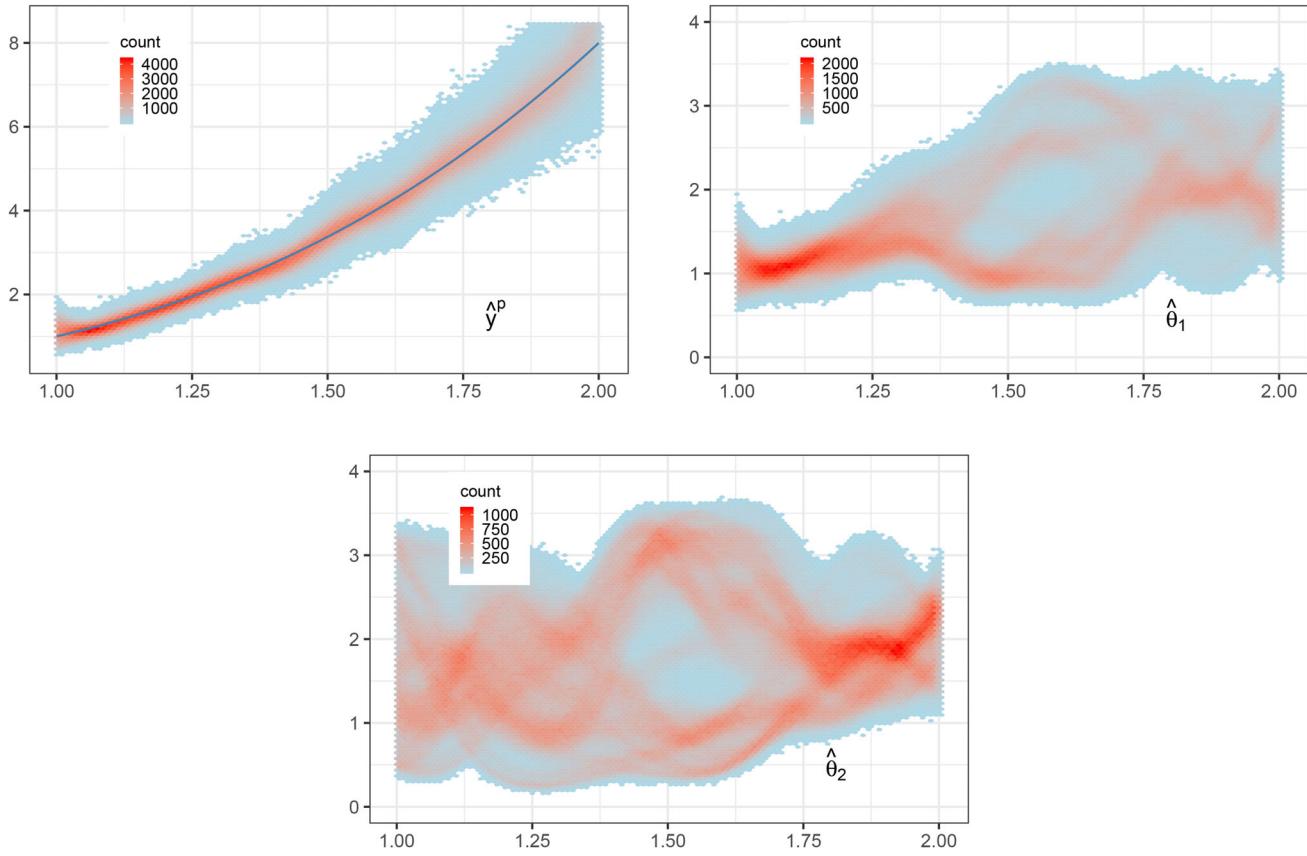


Figure 1. Posterior distributions from the existing Bayesian method for one simulated dataset from Simulation 4. Each panel shows the heat map of the posterior distribution of the underlying function sampled by MCMC after burn-in. The three panels correspond to the physical response \hat{y}^p , and two functional calibration parameters $\hat{\theta}_1, \hat{\theta}_2$, respectively. The solid curve in the first panel represents the true physical response curve.

calibration method to the real dataset from Pourhabib et al. (2015) and compare with the exponential calibration function used in that work.

Now we provide some background of the problem. Buckypaper is a thin sheet of carbon nanotubes. As far as its mechanical properties are concerned, buckypaper is not directly suitable for most applications. To make it useable for practical purposes, one method is to form composites of buckypaper (Tsai et al. 2011), and another approach is to add Poly-vinyl alcohol (PVA) (Wang 2013). In the latter, which yields PVA-treated buckypaper, the goal is to enhance the tensile strength of the final product, measured in terms of Young's modulus. Practitioners want to understand how the stiffness of the buckypaper, measured in terms of the Young's modulus, is affected by the addition of PVA in the fabrication process in the presence of other noise variables. A standard approach is to conduct a set of physical experiments; that is, fabricate a number of buckypapers with varying amounts of the PVA added, measure the Young's modulus of the resulting buckypaper, and fit a functional relationship between the PVA input and the stiffness output. Because measuring the Young's modulus requires a process that damages the buckypaper under test, the physical experiments are expensive to conduct, both time-wise and cost-wise. Therefore, a computer model based on a finite element approximation has been developed to numerically calculate the Young's modulus of the buckypaper under a given amount of PVA additive and a few specifications of carbon nanotubes (Wang 2013).

Pourhabib et al. (2015) reported that this computer model tends to underestimate the Young's modulus for small amounts of PVA and overestimate the modulus for larger amounts of PVA. Understanding of the physical process suggests that such a mismatch is caused by the assumption made in the simulation that the *effectiveness* of PVA—that is, the percentage of the PVA absorbed in the process—stays unchanged as its amount varies. For the computer model outputs to better match the physical experiment outcomes, Pourhabib et al. (2015) considered a modified computer model that includes the effectiveness as a calibration parameter. To determine the value of this parameter is a case of functional calibration because the effectiveness depends on the PVA amount, which is the control variable.

The total number of physical data points is 17. To train the emulator model, we used 150 simulated data points from the finite element model. To remove the model bias of the computer model, a constant was subtracted from the physical data such that the physical data and the simulated data have equal average response over their common range of the control variable. Then the six methods used in the previous section were applied. The fitted calibration functions $\theta(\cdot)$ for various methods are presented in Figure 2. The Const method and the laGP method have constant calibration parameter and their results are not plotted. All fitted functions show that the effectiveness is a roughly decreasing function of the PVA amount in the range $[0.7, 1.1]$, while the two parametric models indicate steeper decrease. The Param-Quad, RKHS-Cubic and the Bayesian method also show

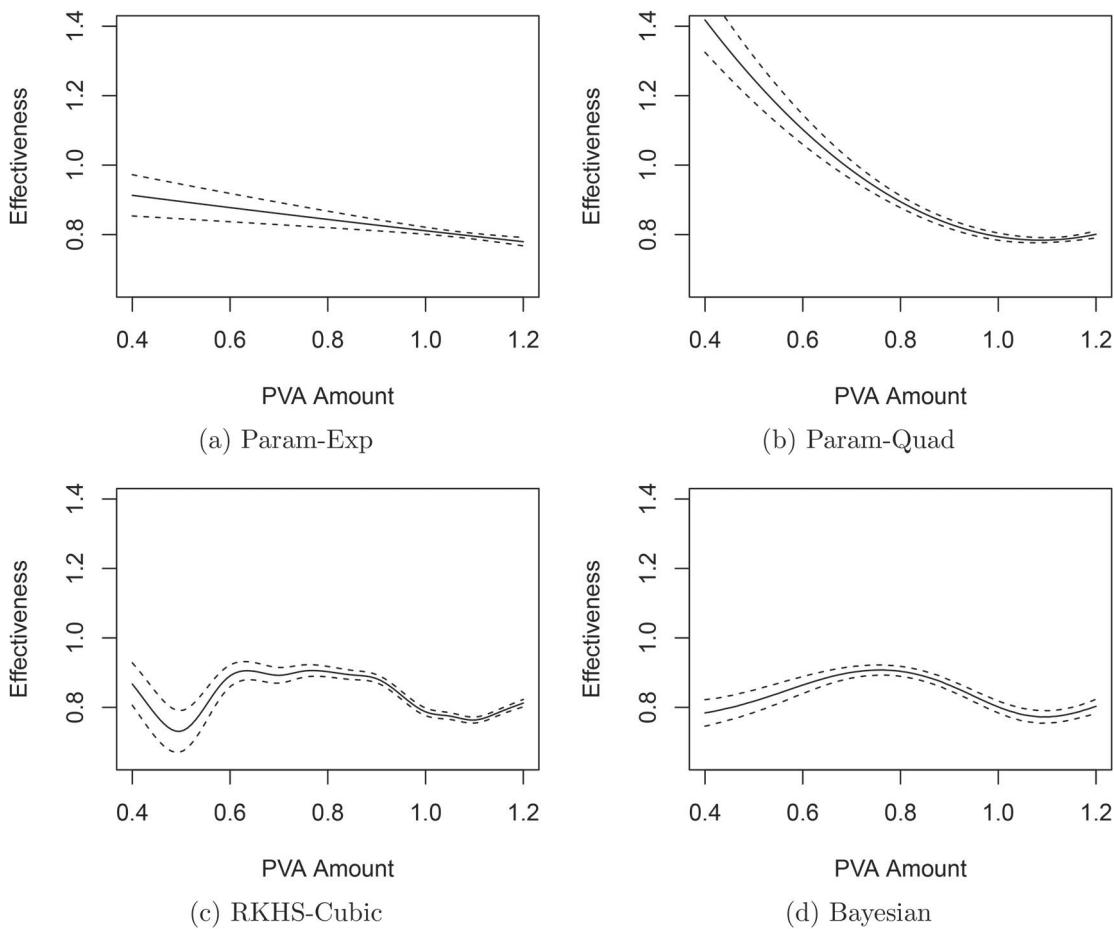


Figure 2. Comparison of four methods for fitting the calibration parameter *effectiveness* as a function of PVA amount. In each panel, the solid line represents the fitted calibration function, and the dashed lines represent the uncertainty by one standard deviation from the fitted function.

Table 5. Leave-one-out and leave-two-out cross-validation absolute prediction error (APE) for modulus prediction in buckypaper fabrication. The mean and standard error (in parenthesis) of the APEs is presented for each method.

leave-C-out	Const	Param-Exp	Param-Quad	RKHS-Cubic	Bayesian	laGP
$C = 1$	130.41 (19.37)	127.71 (25.06)	111.83 (17.68)	59.73 (9.33)	77.89 (19.21)	67.88 (15.13)
$C = 2$	129.72 (5.86)	124.79 (7.93)	110.15 (4.74)	60.32 (3.09)	88.24 (7.61)	80.16 (4.72)

increase of the effectiveness in the PVA range [1.1, 1.2]. The two functional calibration methods, RKHS-Cubic and the Bayesian method, yield calibration functions with similar general shape, while the result of RKHS-Cubic exhibits greater local variability.

We also used cross-validation to compare the prediction performance of the methods, using the absolute prediction error (APE) as the metric for evaluation. Suppose y_i^P is the observed response of the i th observation of the physical data, and y_i^{CV} is the predicted response using the model fitted by leaving out the i th observation. The corresponding cross-validated APE is computed as $APE_i = |y_i^P - y_i^{CV}|$. From Table 5, our RKHS-Cubic clearly has the smallest average leave-one-out cross-validated APE. Because we only have seventeen physical data points in total for leave-one-out cross-validation, the standard errors are too big to claim statistical significance. We then performed a leave-two-out cross-validation which randomly leaves two observations out and computed the cross-validated APE for

each leave-out observation similar to above. The leave-two-out process was repeated 100 times and the results are presented in Table 5. The substantially smaller standard errors returned by the leave-two-out cross-validation indicates that the better performance of RKHS-Cubic over competing methods is statistically significant.

8. Conclusion

The existence of different types of dependency among attributes in complex systems is a well-known fact. In the context of computer experiments, this article considers the dependency between the calibration parameter and control variables through a functional relationship. In this context, unique challenges are present, such as how to model that dependency, how to obtain predictions, what the theoretical properties of the prediction procedures are, and how to quantify the uncertainty of the predictions. This article develops a frequentist approach

to answer such questions. While existing Bayesian methods require to fully specify a probabilistic model and the computation is based on Markov chain Monte Carlo sampling, our frequentist approach does not rely on a concrete probabilistic model and is based on optimization. We have showed that the frequentist approach performs competitively in finite sample for both prediction and uncertainty quantification.

The subject of functional calibration of computer models is clearly widely open for future research. Both frequentist and Bayesian approaches need further development to apply to more sophisticated physical systems. This article provides the first set of asymptotic results on functional calibration of computer models such as consistency and rates of convergence. These results only apply to the frequentist approach. Whether and under what conditions the Bayesian approach has frequentist asymptotic properties are unclear. Asymptotic coverage property of confidence intervals for both frequentist and Bayesian approaches is unstudied and still an important research problem.

Supplementary Materials

The supplementary materials contain the proofs of Theorems 1–4.

Acknowledgment

We are grateful to the associate editor and reviewers for many valuable comments which helped significantly improve previous versions of the article.

Funding

Shiyuan He's work was supported by NSFC Project 11801561. Ding's work was partially supported by NSF grants CMMI-1545038, IIS-1849085, CCF-1934904. Huang's work was partially supported by NSF grants DMS-1208952, IIS-1900990, CCF-1956219. The first two authors, Tuo and He, made equal contributions to the article.

ORCID

Shiyuan He  <http://orcid.org/0000-0003-2593-4566>
Arash Pourhabib  <http://orcid.org/0000-0002-9394-8196>

References

Atamturktur, S., Hegenderfer, J., Williams, B., Egeberg, M., Lebensohn, R. A., and Unal, C. (2015), “A Resource Allocation Framework for Experiment-Based Validation of Numerical Models,” *Mechanics of Advanced Materials and Structures*, 22, 641–654. [2]

Bayarri, M., Berger, J., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, J., Parthasarathy, R., Paulo, R., Sacks, J., and Walsh, D. (2007), “Computer Model Validation With Functional Output,” *The Annals of Statistics*, 35, 1874–1906. [1,2]

Bayarri, M., Berger, J., Paulo, R., Sacks, J., Cafeo, J., Cavendish, J., Lin, C., and Tu, J. (2007), “A Framework for Validation of Computer Models,” *Technometrics*, 49, 138–154. [1]

— (2007), “A Framework for Validation of Computer Models,” *Technometrics*, 49, 138–154. [1]

Brown, D. A., and Atamturktur, S. (2018), “Nonparametric Functional Calibration of Computer Models,” *Statistica Sinica*, 28, 721–742. [2,4,8,11]

Byrd, R. H., and Nocedal, J. (1989), “A Tool for the Analysis of Quasi-Newton Methods With Application to Unconstrained Minimization,” *SIAM Journal on Numerical Analysis*, 26, 727–739. [6]

Craven, P., and Wahba, G. (1979), “Smoothing Noisy Data With Spline Functions,” *Numerische Mathematik*, 31, 377–403. [6]

Danchin, R. (2005), *Fourier Analysis Methods for pdes*. [5]

Duchon, J. (1977), “Splines Minimizing Rotation-Invariant Semi-Norms in Sobolev Spaces,” in *Constructive Theory of Functions of Several Variables*, eds. W. Schempp and K. Zeller, Berlin: Springer; pp. 85–100. [5]

Edmunds, D., and Triebel, H. (2008), *Function Spaces, Entropy Numbers, Differential Operators*, Cambridge: Cambridge University Press. [5]

Golub, G. H., Heath, M., and Wahba, G. (1979), “Generalized Cross-Validation as a Method for Choosing a Good Ridge Parameter,” *Technometrics*, 21, 215–223. [6]

Gramacy, R. B. (2016), “LAGP: Large-Scale Spatial Modeling Via Local Approximate Gaussian Processes in R,” *Journal of Statistical Software*, 72, 1–46. [9]

Gramacy, R. B., Bingham, D., Holloway, J. P., Grosskopf, M. J., Kuranz, C. C., Rutter, E., Trantham, M., and Drake, R. P. (2015), “Calibrating a Large Computer Experiment Simulating Radiative Shock Hydrodynamics,” *The Annals of Applied Statistics*, 9, 1141–1168. [9]

Gu, C. (2013), *Smoothing Spline ANOVA Models*, New York: Springer. [6,7,8,9]

Gu, M., and Wang, L. (2018), “Scaled Gaussian Stochastic Process for Computer Model Calibration and Prediction,” *SIAM/ASA Journal on Uncertainty Quantification*, 6, 1555–1583. [1]

Han, G., Santner, T., and Rawlinson, J. (2009), “Simultaneous Determination of Tuning and Calibration Parameters for Computer Experiments,” *Technometrics*, 51, 464–474. [1]

Higdon, D., Gattiker, J., Lawrence, E., Jackson, C., Tobis, M., Pratola, M., Habib, S., Heitmann, K., and Price, S. (2013), “Computer Model Calibration Using the Ensemble Kalman Filter,” *Technometrics*, 55, 488–500. [1]

Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008), “Computer Model Calibration Using High-dimensional Output,” *Journal of the American Statistical Association*, 103, 570–583. [1]

Higdon, D., Kennedy, M., Cavendish, J., Cafeo, J., and Ryne, R. (2004), “Combining Field Data and Computer Simulations for Calibration and Prediction,” *SIAM Journal of Scientific Computing*, 26, 448–466. [1]

Johnson, M., Moore, L., and Ylvisaker, D. (1990), “Minimax and Maximin Distance Designs,” *Journal of Statistical Planning and Inference*, 26, 131–148. [5]

Joseph, V., and Melkote, S. (2009), “Statistical Adjustments to Engineering Models,” *Journal of Quality Technology*, 41, 362–375. [1]

Joseph, V. R., and Yan, H. (2015), “Engineering-Driven Statistical Adjustment and Calibration,” *Technometrics*, 57, 257–267. [1]

Kennedy, M., and O'Hagan, A. (2001), “Bayesian Calibration of Computer Models” (with discussion), *Journal of the Royal Statistical Society, Series B*, 63, 425–464. [1]

Nychka, D. (1988), “Bayesian Confidence Intervals for Smoothing Splines,” *Journal of the American Statistical Association*, 83, 1134–1143. [8]

Plumlee, M. (2017), “Bayesian Calibration of Inexact Computer Models,” *Journal of the American Statistical Association*, 112, 1274–1285. [1]

— (2019), “Computer Model Calibration with Confidence and Consistency,” *Journal of the Royal Statistical Society, Series B*, 81, 519–545. [1]

Plumlee, M., Joseph, V. R., and Yang, H. (2016), “Calibrating Functional Parameters in the Ion Channel Models of Cardiac Cells,” *Journal of the American Statistical Association*, 111, 500–509. [2]

Pourhabib, A., Huang, J. Z., Wang, K., Zhang, C., Wang, B., and Ding, Y. (2015), “Modulus Prediction of Buckypaper Based on Multi-Fidelity Analysis Involving Latent Variables,” *IIE Transactions*, 47, 141–152. [1,2,8,11,12]

Rasmussen, C., and Williams, C. (2006), *Gaussian Processes for Machine Learning*. Cambridge, MA: The MIT Press. [9]

Santner, T., Williams, B., and Notz, W. (2003), *The Design and Analysis of Computer Experiments*, New York: Springer. [1,3]

Schölkopf, B., Herbrich, R., and Smola, A. J. (2001), “A Generalized Representer Theorem,” in *Computational Learning Theory*, Berlin: Springer; pp. 416–426. [6]

Stein, M. (1999), *Interpolation of Spatial Data: Some Theory for Kriging*, New York: Springer. [4]

Stone, C. J. (1982), "Optimal Global Rates of Convergence for Nonparametric Regression," *The Annals of Statistics*, 10, 1040–1053. [6]

Tsai, C., Zhang, C., David, J. A., Wang, B., and Liang, R. (2011), "Elastic Property Prediction of Single-Walled Carbon Nanotube Buckypaper/Polymer Nanocomposites: Stochastic Bulk Response Modeling," *Journal of Nanoscience and Nanotechnology*, 11, 2132–2141. [12]

Tuo, R. (2019), "Adjustments to Computer Models Via Projected Kernel Calibration," *SIAM/ASA Journal on Uncertainty Quantification*, 7, 553–578. [1]

Tuo, R., and Wu, C. F. J. (2015), "Efficient Calibration for Imperfect Computer Models," *The Annals of Statistics*, 43, 2331–2352. [1,2,3,5]

— (2016), "A Theoretical Framework for Calibration in Computer Models: Parametrization, Estimation and Convergence Properties," *SIAM/ASA Journal on Uncertainty Quantification*, 4, 767–795. [1,2,3,8]

Utreras, F. I. (1988), "Convergence Rates for Multivariate Smoothing Spline Functions," *Journal of Approximation Theory*, 52, 1–27. [5]

van de Geer, S. (2000), *Empirical Processes in M-Estimation*, Cambridge: Cambridge University Press. [6]

Wahba, G. (1983), "Bayesian 'Confidence Intervals' for the Cross-Validated Smoothing Spline," *Journal of the Royal Statistical Society, Series B*, 45, 133–150. [8]

— (1990), *Spline Models for Observational Data* (Vol. 59), Philadelphia, PA: Society for Industrial and Applied Mathematics. [7]

Wang, K. (2013), "Statistics-enhanced Multistage Process Models for Integrated Design and Manufacturing of Poly(vinyl alcohol) Treated Buckypaper," Ph. D. thesis, Florida State University, Tallahassee, FL. [12]

Wang, Y., Yue, X., Tuo, R., Hunt, J. H., and Shi, J. (2020), "Effective Model Calibration Via Sensible Variable Identification and Adjustment With Application to Composite Fuselage Simulation," *Annals of Applied Statistics*, 14, 1759–1776. [1]

Wendland, H. (2005), *Scattered Data Approximation*, Cambridge: Cambridge: Cambridge University Press. [3,5]

Wong, R. K. W., Storlie, C. B., and Lee, T. C. M. (2017), "A Frequentist Approach to Computer Model Calibration," *Journal of the Royal Statistical Society, Series B*, 79, 635–648. [1,3]

Xie, F., and Xu, Y. (2020), "Bayesian Projected Calibration of Computer Models," *Journal of the American Statistical Association*. doi:10.1080/01621459.2020.1753519. [1]

Zhou, D.-X. (2002), "The Covering Number in Learning Theory," *Journal of Complexity*, 18, 739–767. [5]