A Bayesian framework for functional calibration of expensive computational models through non-isometric matching

Babak Farmanesh, Arash Pourhabib, Balabhaskar Balasundaram, Austin Buchanan[*]

Abstract: We study statistical calibration, i.e., adjusting features of a computational model that are not observable or controllable in its associated physical system. We focus on functional calibration, which arises in many manufacturing processes where the unobservable features, called calibration variables, are a function of the input variables. A major challenge in many applications is that computational models are expensive and can only be evaluated a limited number of times. Furthermore, without making strong assumptions, the calibration variables are not identifiable. We propose Bayesian non-isometric matching calibration (BNMC) that allows calibration of expensive computational models with only a limited number of samples taken from a computational model and its associated physical system. BNMC replaces the computational model with a dynamic Gaussian process (GP) whose parameters are trained in the calibration procedure. To resolve the identifiability issue, we present the calibration problem from a geometric perspective of non-isometric curve to surface matching, which enables us to take advantage of combinatorial optimization techniques to extract necessary information for constructing prior distributions. Our numerical experiments demonstrate that in terms of prediction accuracy BNMC outperforms, or is comparable to, other existing calibration frameworks.

Keywords: Functional calibration, Gaussian process, Generalized minimum spanning tree.

1 Introduction

Experimenting on computational models to understand physical systems has been a popular practice ever since computers became advanced enough to handle complex mathematical models and intense computational procedures (Fang et al., 2005, Santner et al., 2013). This popularity is

[*] School of Industrial Engineering and Management, Oklahoma State University, {babak.farmanesh, arash.pourhabib, baski, buchanan}@okstate.edu

mainly because a computational model can obtain outputs of an experiment in a relatively more cost-effective and timely manner compared to conducting actual experiments in a laboratory. However, one challenge in utilizing computational models is their "adjustment." In fact, computational models usually incorporate features that cannot be observed or measured in physical systems, but must be correctly specified so that the computational model can accurately represent the physical system (Kennedy and O'Hagan, 2001). We refer to these unobservable/unmeasurable features as *calibration variables*, and to the adjustment of their values as the *calibration procedure*. We call the input features which are common between the computational models and the physical systems as *control variables*.

For example, in the fabrication of poly-vinyl alcohol (PVA) treated buckypaper, we are interested in understanding the relationship between the response value, which is the tensile strength, and the control variable, which is the PVA amount (Pourhabib et al., 2015). Here, the calibration variable is the percentage of PVA absorbed, which cannot be measured in the physical system, but is required in the computational model.

Past studies on the calibration problem generally assumed unique values for calibration variables, an approach referred to as *global calibration*, and used different statistical approaches to estimate these values. For instance, Kennedy and O'Hagan (2001), Craig et al. (2001), Reese et al. (2004), Higdon et al. (2004, 2008), Williams et al. (2006), Bayarri et al. (2007), and Goldstein and Rougier (2009) devised various Bayesian models, whereas Loeppky et al. (2006) and Pratola et al. (2013) used maximum likelihood estimation, and Joseph and Melkote (2009) and Han et al. (2009) developed mixed models by combining frequentist and Bayesian methodologies. More recently Tuo and Wu (2015, 2016) developed models based on L_2 distance projection to estimate the true values of the global calibration variables.

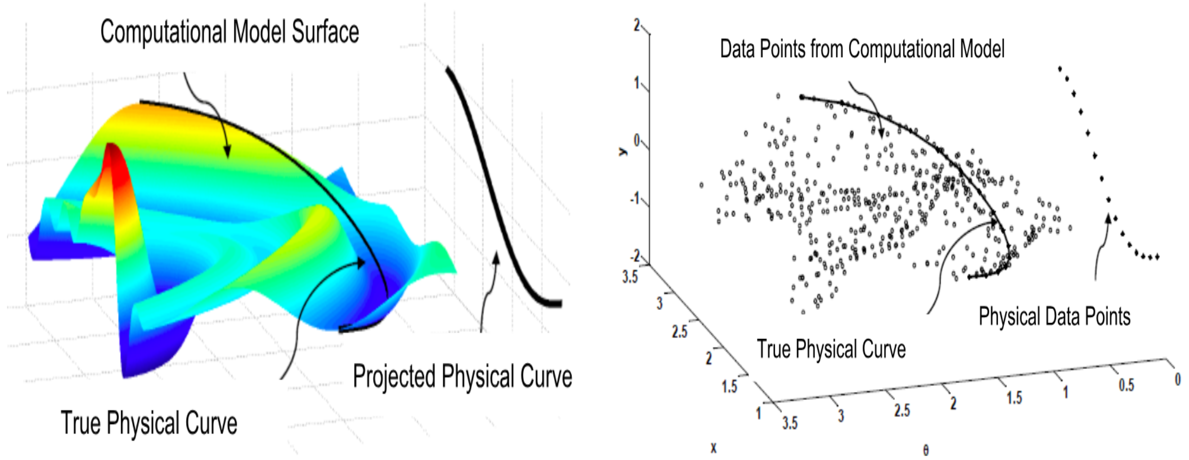
Presently, few studies employ *functional calibration* by assuming that the values of the calibration variables *depend on the control variables*. Pourhabib et al. (2015) showed that, for the buckypaper fabrication problem, an approach that considers a parametric functional relationship between the amount of PVA and the percentage absorbed can outperform the global calibration approach of Kennedy and O'Hagan (2001). Similarly, Xiong et al. (2009) used a simple linear relationship to improve the calibration accuracy in a benchmark thermal challenge problem. Furthermore, non-parametric methods can also be utilized to model functional relationships between the calibration variables and the control variables. Such non-parametric functional relationships have been constructed using Reproducing Kernel Hilbert Spaces (Schölkopf et al., 2001) and Gaussian processes (Rasmussen, 2004) by various authors (Pourhabib et al., 2018, Plumlee et al., 2016, Brown and Atamturktur, 2018).

All the aforementioned studies in functional calibration and most studies in global calibration require computational models that are "cheaply executable." This assumption is required since computational models need to be evaluated thousands of times either to draw samples from a posterior distribution in Bayesian approaches, or to numerically minimize a loss function in other approaches. If the computational model is "expensive," one can obtain a small number of observations from the computational model, then fit a surrogate function based on these random samples, and in the final step, replace the computational model in the calibration procedure with this new surrogate model. However, as discussed in Section 6, this poses a challenge because "static" replacement may result in poor retrieval of the calibration variables.

Another challenge is the identifiability issue: it is difficult to solve the calibration problem in higher dimensional spaces without making additional assumptions about the solution space (Pourhabib et al., 2018). Furthermore, good prediction performance for the response values does not necessarily imply that a method has accurately captured the functional relationship between the calibration and control variables (Tuo and Wu, 2015, Plumlee and Joseph, 2018, Ezzat et al., 2018). This is a significant drawback since, in many applications, understanding the functional relationship between the calibration and control variables is as important as predicting the response values of the system under study.

In this paper, we develop a new framework for the functional calibration of expensive computational models. Unlike conventional surrogate modeling, which replaces the computational model with a static, approximated surface, we employ a "dynamic" Gaussian process (GP) over the computational model. Our GP is dynamic in the sense that the hyper-parameters of the GP's covariance function are trained during the calibration procedure. We simultaneously construct posterior distributions for the hyper-parameters of the GP's covariance function and the calibration variables associated with each of the physical control vectors. In other words, we allow the GP to tune its hyper-parameters in addition to the calibration variables such that the computational model responses become as close as possible to the physical responses.

To tackle the unidentifiability issue in higher-dimensional spaces, we use informative prior distributions. We take advantage of an alternative geometric interpretation of calibration, namely the non-isometric matching of a curve to a surface. We explain this in the case of a single control variable and a single calibration variable. From a geometric perspective, all possible values for the control variable and the physical response constitute a plane curve in the control-response space (see Figure 1a). By contrast, in the computational model, we can specify the values of both the control and the calibration variables. Consequently, all possible values of the control and calibration variables, and the responses of the computational model together form a surface. The plane physical curve we observe in the control-response space is a projection of a space curve in the three-dimensional control-calibration-response space. By nature of projection into a lower-dimensional space, the length of the projected curve is not necessarily the same as the original curve in the three-dimensional space. The projection is therefore *non-isometric* (Bronstein et al., 2003, 2005).



(a) Complete curve and surface

(b) Observed curve and surface by incomplete data

Figure 1: A non-isometric curve to surface matching perspective of functional calibration: The left plot shows the complete surface and curve. In practice, we observe a scatter of data points sampled from the complete curve and surface, which is depicted in the right plot.

The geometric interpretation is due to the nature of the calibration variable in a physical process: for each value of the control variable there exists a (possibly unknown) value for the calibration variable, and these two features determine a single response. Since we do not observe the actual value of the calibration variable in the physical process, we only see a projected curve in the control-response space. Hence, calibration aims to recover the true physical curve, or in other words, determine a non-isometric match of a curve to a surface.

The remainder of this paper is organized as follows. We explain our Bayesian model for handling expensive computational models in Section 2. Section 3 contains a formal description of the calibration problem and its interpretation as a non-isometric curve to surface matching problem. Our graph-theoretic approach to utilizing this geometric perspective to construct informative prior distributions for our Bayesian model is also presented in Section 3. In Section 4, we generalize the idea of a non-isometric curve to surface matching to higher dimensions and introduce integer programming techniques to tackle the problem. The approach presented in Sections 3 and 4 to construct informative prior distributions for our Bayesian model is used to calculate posterior distributions in Section 5. Our experimental results are reported in Section 6, comparing them with previous approaches. Section 7 concludes the paper and presents paths for future research.

2 General setting: a Bayesian model for calibration

Consider a physical system that operates according to a set of (possibly unknown) physical laws. In this system, there is a functional relationship between a group of features and the response (output). We call those features of the system that can be measured and specified as inputs of the physical system as *control variables*, and denote the vector of these variables by $\mathbf{x} \in \mathbb{R}^{d^x}$. We assume that we obtain data for the physical system by conducting *physical experiments*: once the control variables are set (either observed or specified) in the physical system, the physical process $\mathcal{F}^p$ generates a real-valued response y^p, that is $y^p = \mathcal{F}^p(\mathbf{x})$.

Although the response is a function of *all* features of the physical system, we write y^p explicitly as a function of $\mathbf{x}$ as the rest of the features are hard to measure or control, and hence we have no control over them in the physical system. We call such features *calibration variables* and categorize them into the following two groups: (i) *global calibration variables*, which have unique values regardless of the values of the control variables, and (ii) *functional calibration variables*, which are functions of control variables.

We denote the vector of global calibration variables by $\boldsymbol{\psi} \in \mathbb{R}^{d^\psi}$ and the vector of functional calibration variables by $\boldsymbol{\theta} \in \mathbb{R}^{d^\theta}$. We also denote the function that maps $\mathbf{x}$ to the k^{th} element of $\boldsymbol{\theta}$, i.e., θ_k, by $\mathcal{F}^\theta_k$ and the vector of all these functions by $\mathcal{F}^\theta = [\mathcal{F}^\theta_1, \dots, \mathcal{F}^\theta_{d^\theta}]^\top$. With a slight abuse of notation, we denote the vector map from $\mathbf{x}$ to $\boldsymbol{\theta}$ using the vector of functions $\mathcal{F}^{\theta}$ as $\boldsymbol{\theta} = \mathcal{F}^{\theta}(\mathbf{x})$.

Suppose we have a computational model constructed according to the laws governing the physical system. Similar to the physical system, the response of our computational model is determined by the interactions between the control and the calibration variables. However, in a computational model we can set the values of all $\mathbf{x}$, $\boldsymbol{\theta}$, and $\boldsymbol{\psi}$ arbitrarily within their respective domains. That is because, unlike physical experiments, there are no constraints on measuring or specifying control or calibration variables in a computational model. If we denote the computational process as $\mathcal{F}^s$, then the response of the computational model can be written as,

$$y^s = \mathcal{F}^s(\mathbf{x}, \boldsymbol{\psi}, \boldsymbol{\theta}). \tag{1}$$

We refer to obtaining a value for y^s, given a combination of $\mathbf{x}$, $\boldsymbol{\psi}$, and $\boldsymbol{\theta}$ in the computational model, as a *computer experiment.*

The goal of *calibration* is to adjust the variables $\boldsymbol{\psi}$ and $\boldsymbol{\theta}$ such that the computational model represents the physical system in the sense that the computational model can predict the physical response at any input location $\mathbf{x}^*$.

Mathematically, calibration can be viewed as the estimation of vectors $\mathcal{F}^{\theta}$ and $\boldsymbol{\psi}$ such that, for any given $\mathbf{x}^*$, the function $\mathcal{F}^s : \mathbb{R}^{d^x} \times \mathbb{R}^{d^{\psi}} \times \mathbb{R}^{d^{\theta}} \longrightarrow \mathbb{R}$ generates a response close to y^{p^*} up to an error ϵ^*, i.e.,

$$y^{p^*} = \mathcal{F}^s(\mathbf{x}^*, \boldsymbol{\psi}, \mathcal{F}^{\theta}(\mathbf{x}^*)) + \epsilon^*, \tag{2}$$

where the error ϵ^* captures the measurement error and the discrepancy between the physical and computation model.

To estimate $\boldsymbol{\psi}$ and $\mathcal{F}^{\theta}$ in (2) we initially obtain m responses from $\mathcal{F}^p$ at a set of physical system inputs $\{\mathbf{x}_1^p, \ldots, \mathbf{x}_m^p\}$ to create a dataset P corresponding to that physical system,

$$P := \left\{ p_i = (\mathbf{x}_i^p, y_i^p) \;\middle|\; \mathbf{x}_i^p \in \mathbb{R}^{d^x}, y_i^p \in \mathbb{R}, i \in \{1, 2, \ldots, m\} \right\}.$$

We also create the counterpart of P in the computational model, i.e., the computational dataset as

$$S := \left\{ s_j = \left(\mathbf{x}_j^s, \boldsymbol{\psi}_j^s, \boldsymbol{\theta}_j^s, y_j^s \right) \;\middle|\; \mathbf{x}_j^s \in \mathbb{R}^{d^x}, \boldsymbol{\psi}_j^s \in \mathbb{R}^{d^\psi}, \boldsymbol{\theta}_j^s \in \mathbb{R}^{d^\theta}, j \in \{1, 2, \dots, n\} \right\},$$

based on a set of computational model inputs $\{(\mathbf{x}_1^s, \boldsymbol{\psi}_1^s, \boldsymbol{\theta}_1^s), \dots, (\mathbf{x}_n^s, \boldsymbol{\psi}_n^s, \boldsymbol{\theta}_n^s)\}$. We assume the sets of physical system inputs and the computational model inputs are given. For a discussion of how to select the inputs we refer the reader to the paper by Ezzat et al. (2018).

Let $\boldsymbol{\theta}_i^p = \mathcal{F}^\theta(\mathbf{x}_i^p)$ and $\boldsymbol{\psi}^p$ denote the true values of the calibration variables and assume that the errors are independent and have identical normal distribution with zero mean and constant variance. Therefore, we obtain the calibration model as

$$y_i^p = \mathcal{F}^s(\mathbf{x}_i^p, \boldsymbol{\psi}^p, \boldsymbol{\theta}_i^p) + \epsilon_i^p, \text{ where } \epsilon_i^p \sim \mathcal{N}(0, \sigma^2), \quad \forall i \in \{1, 2, \dots, m\}. \tag{3}$$

Remark 1. If we remove the functional calibration variable $\boldsymbol{\theta}_i^p$ from equation (3), we get a simplified version of the global calibration model proposed in (Kennedy and O'Hagan, 2001). In fact, Kennedy and O'Hagan (2001) assume $y_i^p = \mathcal{F}^s(\mathbf{x}_i^p, \boldsymbol{\psi}^p) + \delta(\mathbf{x}_i^p) + \epsilon_i^p$, where $\delta(\cdot)$ is a GP independent from $\mathcal{F}^s$, which characterizes all the discrepancy between the computational model and the physical system due to assumptions made in building the computational model. However, because in this study we focus on computational models with a limited number of data points, we do not include a separate discrepancy term $\delta(\cdot)$, which entails introducing a set of additional parameters and would make the estimation procedure unstable. Therefore, as we utilize a dynamic GP to minimize the overall discrepancy, we choose to use ϵ_i^p to represent not only the measurement error in the physical system but also the discrepancy between the computational model and the physical system. As such, our assumptions are similar to those made by Brown and Atamturktur (2018). In Appendix E we validate these assumptions on the datasets used in this study. The reader can refer to the discussion by Tuo and Wu (2016) for a frequentist interpretation of the model proposed by Kennedy and O'Hagan (2001).

Note that applying Bayesian statistics to construct posterior distributions for parameters of model (3), i.e., $\boldsymbol{\psi}^p, \sigma^2$ and $\boldsymbol{\theta}_i^p$, requires a large number of evaluations of $\mathcal{F}^s$, which is not practical for expensive computational models. Therefore, we assume $\mathcal{F}^s$ is a GP with a constant mean (Rasmussen, 2004), i.e., $\mathcal{F}^s \sim \mathcal{GP}(0, \mathcal{K}(\cdot,\cdot))$, where $\mathcal{K}(\cdot,\cdot)$ is a covariance function. We further assume that the overall average of the responses from the computational model has been subtracted from each output, and as such we use a GP with mean zero. Here we use the squared exponential kernel function as the choice of the covariance function,

$$\mathcal{K}(\mathbf{z}, \mathbf{z}') = \gamma \exp(-(\mathbf{z} - \mathbf{z}')^\top \mathbf{L}(\mathbf{z} - \mathbf{z}')), \tag{4}$$

where γ is the magnitude parameter and $\mathbf{L}$ is a diagonal matrix of the length-scale parameters. We denote the vector of the diagonal elements of $\mathbf{L}$ by $\boldsymbol{\ell}$.

Subsequently, we can obtain the likelihood of model (3) by the GP distribution defined on $\mathcal{F}^s$ as a multivariate normal distribution,

$$\mathbf{y}^p \mid \mathbf{X}^p, \boldsymbol{\Theta}^p, \boldsymbol{\psi}^p, \boldsymbol{\ell}, \gamma, \sigma^2 \sim \mathcal{N}(0, \boldsymbol{\Sigma} + \sigma^2 \mathbf{I}_m), \tag{5}$$

where $\mathbf{y}^p = [y_1^p, \ldots, y_m^p]^\top$ is the vector of physical responses, $\mathbf{X}^p = [\mathbf{x}_1^p, \ldots, \mathbf{x}_m^p]^\top$ and $\boldsymbol{\Theta}^p = [\boldsymbol{\theta}_1^p, \ldots, \boldsymbol{\theta}_m^p]^\top$ are matrices of size $m \times d^x$ and $m \times d^\theta$ respectively, and $\boldsymbol{\Sigma}$ is the $m \times m$ covariance matrix whose elements are calculated by covariance function (4) with $[\mathbf{x}_i^{p^\top}, \boldsymbol{\theta}_i^{p^\top}, \boldsymbol{\psi}^{p^\top}]^\top$ as input vectors with length $(d^x + d^\theta + d^\psi)$.

Remark 2. Although in the process of deriving likelihood (5), we consider $\boldsymbol{\psi}^p$ and the columns of $\boldsymbol{\Theta}^p$ as the input variables of model (3), we do not know the values of these input variables, and we intend to estimate them. Therefore, in order to distinguish the calibration variables $\boldsymbol{\psi}$ and $\boldsymbol{\theta}$, in (1) from the parameters in model (3), we refer to $\boldsymbol{\Theta}^p$ and $\boldsymbol{\psi}^p$ as *calibration parameters.*

We can estimate the calibration parameters of model (3), the parameters of covariance function (4), and the variance of error, using Bayesian statistics with the posterior distribution,

$$\pi(\boldsymbol{\Theta}^p, \boldsymbol{\psi}^p, \boldsymbol{\ell}, \gamma, \sigma^2 \mid \mathbf{y}^p, \mathbf{X}^p) \propto \pi(\mathbf{y}^p \mid \mathbf{X}^p, \boldsymbol{\Theta}^p, \boldsymbol{\psi}^p, \boldsymbol{\ell}, \gamma, \sigma^2)\pi(\boldsymbol{\Theta}^p)\pi(\boldsymbol{\psi}^p)\pi(\boldsymbol{\ell})\pi(\gamma)\pi(\sigma^2). \tag{6}$$

The Bayesian model (6) would be completed by specifying prior distributions for the parameters $\pi(\boldsymbol{\Theta}^p)$, $\pi(\boldsymbol{\psi}^p)$, $\pi(\boldsymbol{\ell})$, $\pi(\gamma)$, and $\pi(\sigma^2)$. However, our model suffers from unidentifiablity in the absence of informative priors due to the high-dimensionality of the parameter space. Therefore, in Section 5 we present graph-theoretic approaches that help construct informative priors for the calibration parameters $\boldsymbol{\Theta}^p$ and $\boldsymbol{\psi}^p$.

Note that the replacement of $\mathcal{F}^s$ by $\mathcal{GP}(0, \mathcal{K}(\cdot,\cdot))$ does not constitute a surrogate modeling approach, wherein the computational model is replaced by a *fixed* surrogate surface, which is in turn trained based on a set of limited samples drawn from the computational model prior to any calibration procedure. Our approach is fundamentally different from surrogate modeling, since building and training the GP is a part of the calibration process.

3 Calibration as non-isometric matching: a special case

We explain in this section how the calibration problem can be viewed as a non-isometric curve to surface matching problem for the special case where $\mathbf{x} \in \mathbb{R}$, $\boldsymbol{\theta} \in \mathbb{R}$, and $\boldsymbol{\psi} \in \varnothing$, which means both control and calibration variables are one-dimensional and no global calibration variable exists. From a geometric perspective, all the possible values for $\mathbf{x}$ and $\mathcal{F}^p(\mathbf{x})$ constitute the curve $(\mathbf{x}, \mathcal{F}^p(\mathbf{x}))$ in a two-dimensional space. In the computational model, however, we can specify the values of both $\mathbf{x}$ and $\boldsymbol{\theta}$. Consequently, all the possible values of $\mathbf{x}$, $\boldsymbol{\theta}$, and $\mathcal{F}^s(\mathbf{x}, \boldsymbol{\theta})$ together form a surface $(\mathbf{x}, \boldsymbol{\theta}, \mathcal{F}^s(\mathbf{x}, \boldsymbol{\theta}))$ in a three-dimensional space. As we noted in Section 1, the true physical curve lies on the three-dimensional computational model surface, i.e., $(\mathbf{x}, \mathcal{F}^\theta(\mathbf{x}), \mathcal{F}^p(\mathbf{x}))$. However, since we do not observe the actual values of the calibration variables in the physical process, we only see a projected curve in $\mathbf{x} - y$ space (see Figure (1a)). Hence, the calibration problem is to recover the true physical curve, or, in other words, determine a non-isometric match of a curve to a surface.

As mentioned earlier, the non-isometry is due to the fact that the curve $(\mathbf{x}, \mathcal{F}^\theta(\mathbf{x}), \mathcal{F}^p(\mathbf{x}))$ on the three-dimensional $\mathbf{x} - \boldsymbol{\theta} - y$ space has a different length than the projected curve $(\mathbf{x}, 0, \mathcal{F}^p(\mathbf{x}))$ on a two-dimensional $\mathbf{x} - y$ space. Therefore, this is, in principle, different from isometric matching problems (Gruen and Akca, 2005, Bronstein et al., 2005, Baltsavias et al., 2008).

In practice we only have the finite physical system dataset P along with a finite computational model dataset S, as we do not observe a complete curve or surface. Ideally, the points in P lie on the projected curve that we observe, and the points in S lie on the computational model surface (see Figure 1b). Hence, what we observe is incomplete data, and we aim to match non-isometrically an incomplete curve to an incomplete surface, which is equivalent to solving the calibration problem.

This geometric perspective motivates us to view the problem through a combinatorial lens and model the problem using graph-theoretic approaches. Our graph-based solution to the non-isometric curve to surface matching problem provides us with a set of computational model data points, which carry information about the calibration parameters. We call this set of computational data points *anchor points*. These anchor points will then be used in Section 5 to construct prior distributions for our Bayesian model.

We seek to identify a set of anchor points among the computational data points that are "close" to the points on the true physical curve. In other words, the anchor points are positioned such that the true physical curve passes through the neighborhoods of those points. We want the anchor points to satisfy two desirable properties: (i) the computational model response should be close to the physical response for a given input $\mathbf{x}$; and (ii) the calibration parameter values for two consecutive anchor points should be close to each other. The former drives our method to identify the anchor points that have similar responses to that of the physical system, and the latter aims to encourage the smoothness of the physical curve.

Note that we are only interested in identifying these "optimal" anchor points that provide us with information about the true physical curve to be used in our prior distributions, and not the true physical curve itself. However, one could also directly use the selected anchor points to approximate the true physical curve via interpolation. Given our focus on expensive computational models wherein the number of computational model data points is limited, such an approximation of the true physical curve may not be accurate. In the next section, we formally define and address the problem of finding anchor points with the desired properties using a graph-theoretic approach for the special case when $\mathbf{x} \in \mathbb{R}$, $\boldsymbol{\theta} \in \mathbb{R}$, and $\boldsymbol{\psi} \in \varnothing$.

3.1 A graph-theoretic approach for finding anchor points

Without loss of generality, we assume that all the data points in the physical system and the computational model datasets are strictly ordered such that $\mathbf{x}_i^p < \mathbf{x}_{i+1}^p$, for all $i \in \{1, 2, \ldots, m-1\}$, and $\mathbf{x}_j^s < \mathbf{x}_{j+1}^s$, for all $j \in \{1, 2, \ldots, n-1\}$. We construct an edge-weighted directed graph $G = (V, E)$ with vertex set $V := \{0, 1, 2, \ldots, n+1\}$, and the edge set E described in equation (8) below. The vertices in $V^0 := \{1, 2, \ldots, n\}$ correspond to the computational model data points in S. We refer to G as the *calibration digraph*.

Recall that, intuitively, the objective of calibration is to minimize the difference between the outputs of the physical system and the corresponding computational model. As such, the first step is to find control variables that are similar in both settings, the physical experiments and the computer experiments. Therefore, we first group the control variables in the computational model based on their distance to the control variables in the physical experiments. We partition V^0 into m clusters $C_1, \ldots, C_m$ as follows: any vertex $j \in V^0$ corresponding to data point $s_j \in S$ is assigned to a unique cluster C_i by the following formula:

$$j \in C_i \iff i = \min \left\{ \underset{\ell \in \{1,2,\ldots,m\}}{\arg\min} \{||\mathbf{x}_\ell^p - \mathbf{x}_j^s||_2\} \right\}. \tag{7}$$

If the inner minimum in (7) is not unique, then the outer minimum is used to break the tie by choosing the smallest index. As a consequence, each cluster C_i is in 1-to-1 correspondence with the i^{th} physical data point. This choice of tie-breaker is easy to implement and establishes a mechanism for consistent assignment of points to a cluster. We can now describe the set of directed edges E as

$$E := \bigcup_{i=1}^{m-1} \{(u,v) \mid u \in C_i, v \in C_{i+1}\} \bigcup \{(0,u) \mid u \in C_1\} \bigcup \{(u, n+1) \mid u \in C_m\}. \tag{8}$$

This construction is illustrated in Figure 2.

The final critical step is to assign a weight w_{uv} to each edge $(u,v) \in E$. Consider two consecutive clusters C_i and C_{i+1} and vertices $u \in C_i$ and $v \in C_{i+1}$. Define w_{uv} as

$$w_{uv} := |y_u^s - y_u^p| + \lambda ||\boldsymbol{\theta}_u^s - \boldsymbol{\theta}_v^s||_2, \tag{9}$$

where $\lambda > 0$ is a scaling parameter. The weights of edges that leave vertex 0 or enter vertex $n+1$ are identically zero. The edge-weight for any edge between two consecutive clusters i and $i+1$ consists of two parts: the first part $|y_u^s - y_u^p|$ represents the difference between the model response and physical response; the second part $||\boldsymbol{\theta}_u^s - \boldsymbol{\theta}_v^s||_2$ represents the difference between the calibration parameters of i and $i+1$. On this digraph G with the given edge-weights, we intend to solve the shortest path problem from origin vertex 0 to destination vertex $n+1$. Every path from vertex 0 to vertex $n+1$ in G has exactly $m+1$ edges by construction. Suppose 0-v_1-v_2-$\cdots$-v_m-$(n+1)$ is the shortest path identified. Then, those points in S corresponding to $\{v_1, \ldots, v_m\}$ serve as the anchor points. The edge-weights quantify the proximity of the physical and computer experiment outputs

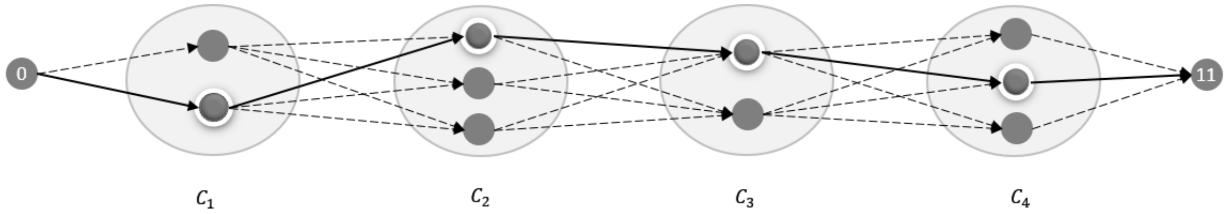


Figure 2: Illustration of the calibration digraph for the case where $m = 4$ and $n = 10$. The vertices represent data points from the computational model and the clusters C_1 through C_m correspond to physical system data points. Vertices denoted by dark circles with a white border represent the anchor points, and the solid arrows identify the edges in the shortest path found.

and difference between the calibration parameters to minimize erratic changes.

Lemma 1. *Calibration digraph $G = (V, E)$ is acyclic with a topological ordering $\langle 0, 1, \ldots, n, n+1 \rangle$.*

Proof. See Appendix B for proofs. □

Since G is a directed acyclic graph, or DAG for short, we can solve the shortest path problem using an $\mathcal{O}(|E|)$ algorithm that scans outgoing edges from each vertex in the topological order and updates distance-labels as needed (Bellman, 1958, Lawler, 1976).

4 Generalization of non-isometric matching to higher dimensions

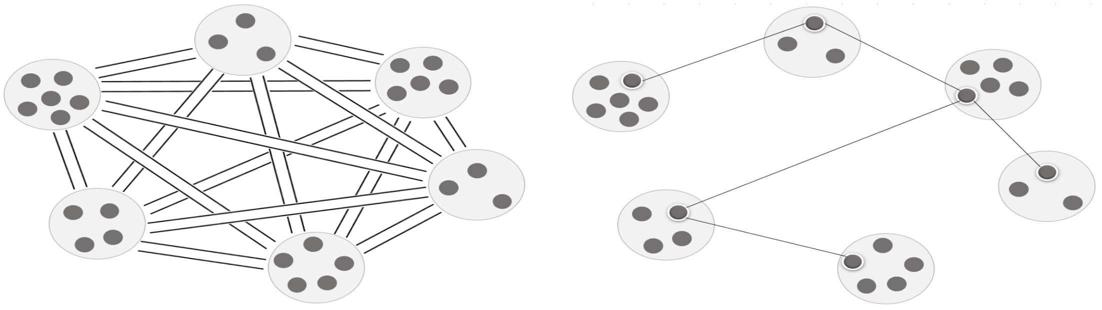
Section 3 introduced the curve to surface matching interpretation of calibration with $\mathbf{x} \in \mathbb{R}$ and $\boldsymbol{\theta} \in \mathbb{R}$. This special case allowed us to develop a graph-theoretic approach for anchor point selection that admitted a fast $\mathcal{O}(|E|)$ algorithm. The geometric perspective can be generalized to arbitrary dimensions as a hyper-curve to hyper-surface matching problem. However, in the general setting, there is no straightforward extension of the directed acyclic graph model. Recall that the model hinges on the natural ordering of the computational and the physical data points on the real line, which does not exist in higher dimensions. So, in this section we introduce a different calibration graph model and an associated combinatorial optimization problem to find the anchor points in an arbitrary dimension. As with the special case, the anchor points will subsequently be used in Section 5 to construct prior distributions for our Bayesian model.

For the general case, we construct a *calibration graph* $G = (V, E)$ that is undirected and edge-weighted, where $V = \{1, 2, \ldots, n\}$ corresponds to the n computational data points. We partition V into m clusters, $C_1, \ldots, C_m$, in correspondence with the m physical data points and assign vertex j to a cluster C_i by the same rule in equation (7). The graph G is a complete m-partite graph with partitions $C_1, \ldots, C_m$, i.e., distinct vertices are adjacent if and only if they belong to different partitions. The edge set can be described formally as $E \coloneqq \bigcup_{i=1}^{m-1} \bigcup_{\ell=i+1}^{m} \{\{u, v\} \mid u \in C_i, v \in C_\ell\}$. Figure 3a illustrates this construction.

Finally, before defining the edge weights, we introduce two required concepts. The *calibration vector* of data point s_j is given by $[\boldsymbol{\theta}_j^{s^\top}, \boldsymbol{\psi}_j^{s^\top}]^\top$. We assign the weight w_e to the edge $e = \{u, v\} \in E$, where $u \in C_i$ and $v \in C_\ell$, by

$$w_e \coloneqq \begin{cases} |y_u^s - y_i^p| + |y_v^s - y_\ell^p| + \lambda ||[\boldsymbol{\theta}_u^{s^\top}, \boldsymbol{\psi}_u^{s^\top}]^\top - [\boldsymbol{\theta}_v^{s^\top}, \boldsymbol{\psi}_v^{s^\top}]^\top||_2 & \text{if } ||\mathbf{x}_u^s - \mathbf{x}_v^s||_2 \leq r \\ |y_u^s - y_i^p| + |y_v^s - y_\ell^p| + M||\mathbf{x}_u^s - \mathbf{x}_v^s||_2 & \text{if } ||\mathbf{x}_u^s - \mathbf{x}_v^s||_2 > r, \end{cases} \tag{10}$$

where λ is a scaling parameter and M is a sufficiently large number used to penalize the computational data points that are far from each other. Note that the weights assigned in (10) extend the idea behind equation (9). Here, the edge weight between vertices $u \in C_i$ and $v \in C_\ell$, where s_u and s_v are *neighbors* (that is, the Euclidean distance between their control vectors is smaller than a predefined radius r), consists of two parts, similar to (9): the first part measures the distance between each vertex's response and the physical system response associated with the cluster to which it belongs, i.e., $|y_u^s - y_i^p|$ and $|y_v^s - y_\ell^p|$; the second part measures the distance between the corresponding calibration vectors, i.e., $||[\boldsymbol{\theta}_u^{s^\top}, \boldsymbol{\psi}_u^{s^\top}]^\top - [\boldsymbol{\theta}_v^{s^\top}, \boldsymbol{\psi}_v^{s^\top}]^\top||_2$. Let E_1 denote the set of all edges that join vertex pairs corresponding to control vectors that are at most Euclidean distance r apart. The remainder of the edges, $E_2 = E \setminus E_1$, correspond to edges between computational data points that are not close enough, and we assign relatively large weights to these edges by setting M to a large value. Furthermore, the weight on such edges increases as the distance between the control vectors of the end points increases.



(a) Calibration graph (b) A generalized minimum spanning tree

Figure 3: (a) A calibration graph where each black circle represents a vertex and each two parallel lines represent edges between vertices of two clusters. (b) A generalized spanning tree in the calibration graph.

To identify the “optimal” anchor vertices from this calibration graph, we find a minimum weight tree that contains exactly one vertex from each cluster. In the optimization literature, this problem is known as the *generalized minimum spanning tree* (GMST) problem (Myung et al., 1995). By our construction of the edge weights, a GMST will tend to include edges in E_1 as they are lighter. However, if no GMST exists that only uses edges in E_1, it will be forced to include edges in E_2.

4.1 Integer programming approaches to the GMST problem

The GMST problem was introduced by Myung et al. (1995), who showed that it is NP-hard and does not admit a polynomial-time constant-factor approximation algorithm unless P = NP. Various authors have developed and analyzed integer programming (IP) formulations for this problem and the strength of the associated linear programming (LP) relaxations (Myung et al., 1995, Feremans et al., 2002, Pop, 2004, Pop et al., 2006). Strong formulations, which correspond to tight LP relaxations, are desirable in a branch-and-bound algorithm as they produce tighter bounds that can be helpful in pruning the search tree. We employ two such strong formulations with tight LP relaxations for solving the anchor point selection problem in arbitrary dimension.

The class of formulations that were first introduced by Myung et al. (1995) employs exponentially many constraints and are analogous to the cutset and subtour elimination formulations of the traveling salesman problem and the minimum spanning tree problem (Bertsimas and Weismantel, 2005). Feremans et al. (2002) showed that strengthening a subtour elimination formulation of a more general variant of the GMST problem is among the strongest in terms of the tightness of the

LP relaxation. We use this formulation, which Feremans et al. (2002) call the *directed cluster subpacking* (DCSUB) formulation, in our computational experiments. This formulation and additional explanation are provided in Appendix C.

Because of the presence of exponentially many constraints, a direct implementation of the entire DCSUB formulation is impractical even for small scale problems. Nonetheless, a delayed constraint generation approach could be effective in practice (Buchanan et al., 2015, Moradi and Balasundaram, 2018, Lu et al., 2018). This approach starts by relaxing the formulation by omitting a subset of the constraints (typically those that are exponentially many in number). During the normal progress of an LP relaxation based branch-and-bound algorithm to solve the relaxed IP, whenever an integral solution is detected at some node of the search tree, it is necessary to verify if a constraint that violates this solution exists among those constraints that were excluded. If so, we solve the model at that node again after adding the violated constraints back; otherwise, we continue to branch as usual, thus ensuring the overall correctness of the algorithm. An effective implementation of such an algorithm is possible using the "lazy cut" feature available in most state-of-the-art IP solvers as long as the identification of the violated constraints can be accomplished quickly.

The second type of formulation we use in our computational experiments is based on the classical multi-commodity network flow (MCF) formulation (Myung et al., 1995, Feremans et al., 2002, Pop, 2004, Pop et al., 2006). The underlying idea of this formulation is to use the flow of a (dummy) "commodity" in the network to trace a path between two vertices by designating one vertex with unit supply for that commodity and the other with unit demand. As the MCF formulation uses only polynomially many constraints and variables, it can be directly implemented and solved using most IP solvers for moderately sized instances. The MCF formulation, which also has a strong LP relaxation, is presented and discussed in greater detail in Appendix C.

Before proceeding to the next section, we point out that although GMST can be applied to any dataset (one- or multi-dimensional), the shortest path model is preferable for one-dimensional problems for the following reasons. First, the shortest path model does not require specifying parameters r and M, and second, it is solvable in time $\mathcal{O}(|V| + |E|)$. By contrast, GMST requires choosing parameters r and M, and it is NP-hard in general. For large scale one-dimensional problems, solving the GMST model may require a sophisticated integer programming approach and may require substantially more computational effort when compared to the shortest path alternative.

5 Prior and posterior distributions

This section describes how the information about the true physical curve carried by the anchor points, found by approaches discussed in Sections 3 and 4, can be used to construct our prior distributions for the calibration parameters. We also expand posterior distribution (6) using the priors specified in this section, and show how we can make predictions at a new control vector $\mathbf{x}^*$.

Suppose $\boldsymbol{\theta}_i^a$ and $\boldsymbol{\psi}_i^a$ are, respectively, the functional and the global calibration vectors of the anchor point associated with the i^{th} physical data point, i.e., the anchor vertex selected from the i^{th} cluster. Recall that the anchor points are selected by minimizing a weighted combination of two measures: a) the difference between the model responses and physical responses, and b) the distance between the corresponding calibration vectors. As such we can utilize those anchor points to build priors for the calibration parameters in a Bayesian model. To account for the uncertainty associated with the selection of the anchor points, we use variance hyperparameters as explained next. Define the matrix $\boldsymbol{\Theta}^a := [\boldsymbol{\theta}_1^a, \ldots, \boldsymbol{\theta}_m^a]^\top$ of size $m \times d^\theta$ and the mean vector $\boldsymbol{\psi}^a := \frac{1}{m}\sum_{i=1}^m \boldsymbol{\psi}_i^a$ of length d^ψ. Note that for the latter we take the average of the global calibration vectors of the anchor points since we assume that the global calibration parameters are constant regardless of the values of the control vectors.

For each component of $\boldsymbol{\psi}^p$, the global calibration parameters, we consider a univariate normal distribution centered at the corresponding element in $\boldsymbol{\psi}^a$ with an unknown variance as the choice of the prior distribution. Therefore, we construct the prior distribution for $\boldsymbol{\psi}^p$ as

$$\boldsymbol{\psi}^p \mid \boldsymbol{\psi}^a, \boldsymbol{\tau}^2 \sim \mathcal{N}(\boldsymbol{\psi}^a, \operatorname{diag}(\boldsymbol{\tau}^2)), \tag{11}$$

where $\boldsymbol{\tau}^2 = [\tau_1^2, \ldots, \tau_{d^\psi}^2]^\top$ is the vector of variances of the normal distribution.

Applying the same procedure for constructing prior distributions for the functional calibration parameters increases the dimension of the parameter space, since we need to define md^θ variance parameters, which are nuisance parameters and not of interest to our model. Therefore, in order to shrink the parameter space, we use the fact that the k^{th} column of the functional calibration parameters $\boldsymbol{\Theta}^p$, i.e. $\boldsymbol{\Theta}^p_k$, is actually a realization of the functional relationship $\mathcal{F}^\theta_k$. Therefore, the k^{th} column of $\boldsymbol{\Theta}^a$, i.e., $\boldsymbol{\Theta}^a_k$, is a rough estimator of this realization. On this basis, we use a single variance parameter for all the elements in $\boldsymbol{\Theta}^p_k$, and construct the prior distribution for $\boldsymbol{\Theta}^p_k$ as

$$\boldsymbol{\Theta}^p_k \mid \boldsymbol{\Theta}^a_k, \nu^2_k \ \sim \mathcal{N}(\boldsymbol{\Theta}^a_k, \nu^2_k \mathbf{I}_m), \tag{12}$$

where ν^2_k is the k^{th} element of the vector of variances $\boldsymbol{\nu}^2$ with length d^θ.

The correctness of the normality assumptions in (11) and (12) is a legitimate concern, because there is no guarantee that the anchor points embrace the true physical curve due to the limited number of observations. However, we only make the normality assumptions in (11) and (12) for constructing the prior distributions, and the Bayesian model will adjust these priors by likelihood (5).

To specify the posterior distribution, we define proper prior distributions for the rest of the parameters. As such, we get:

$$\begin{aligned} \pi(\boldsymbol{\Theta}^p, \boldsymbol{\psi}^p, \boldsymbol{\nu}^2, \boldsymbol{\tau}^2, \boldsymbol{\ell}, \gamma, \sigma^2 \mid \mathbf{y}^p, \mathbf{X}^p, \boldsymbol{\Theta}^a, \boldsymbol{\psi}^a) \propto \\ \pi(\mathbf{y}^p \mid \mathbf{X}^p, \boldsymbol{\Theta}^p, \boldsymbol{\psi}^p, \boldsymbol{\ell}, \gamma, \sigma^2)\pi(\boldsymbol{\Theta}^p \mid \boldsymbol{\Theta}^a, \boldsymbol{\nu}^2)\pi(\boldsymbol{\psi}^p \mid \boldsymbol{\psi}^a, \boldsymbol{\tau}^2)\pi(\boldsymbol{\nu}^2)\pi(\boldsymbol{\tau}^2)\pi(\boldsymbol{\ell})\pi(\gamma)\pi(\sigma^2), \end{aligned} \tag{13}$$

where,

$$\mathbf{y}^p \mid \mathbf{X}^p, \boldsymbol{\Theta}^p, \boldsymbol{\psi}^p, \boldsymbol{\ell}, \gamma, \sigma^2 \sim \mathcal{N}(0, \boldsymbol{\Sigma} + \sigma^2 \mathbf{I}_m).$$

We refer the reader to Appendix A for the exact prior distributions of parameters in (13), and a discussion of how to sample from the posterior distribution.

In order to make predictions at a new control vector $\mathbf{x}^*$, we introduce variables $\boldsymbol{\psi}^p(t)$, $\boldsymbol{\ell}(t)$, $\gamma(t)$, $\sigma^2(t)$, and $\boldsymbol{\Theta}^p(t)$ as t^{th} draws from the posterior distribution (13) after some burn-in period, where $t \in \{1, \dots, T\}$. The first step in the prediction of response y^* is to estimate the associated functional calibration vector of $\mathbf{x}^*$, i.e., $\boldsymbol{\theta}^* = \mathcal{F}^\theta(\mathbf{x}^*)$. We can estimate $\boldsymbol{\theta}^*$ based on each $\boldsymbol{\Theta}^p(t)$, where we denote the t^{th} estimation of $\boldsymbol{\theta}^*$ based on $\boldsymbol{\Theta}^p(t)$ as $\boldsymbol{\theta}^*(t)$. To this end we note that as $\boldsymbol{\Theta}^p_k(t)$ is a vector of estimates of $\mathcal{F}^\theta_k$ at the design locations $\{\mathbf{x}^p_1, \dots, \mathbf{x}^p_m\}$, we can write $\boldsymbol{\Theta}^p_k(t) = [\mathcal{F}^\theta_k(\mathbf{x}^p_1), \dots, \mathcal{F}^\theta_k(\mathbf{x}^p_m)]^\top + [\epsilon^\theta_1, \dots, \epsilon^\theta_m]^\top$, where $[\epsilon^\theta_1, \dots, \epsilon^\theta_m]$ is a vector of the corresponding error terms. The error term appears because $\mathbf{\Theta}^p_k(t)$ does not contain exact evaluations of the function $\mathcal{F}^\theta_k$ but only estimations. The following proposition obtains the mean prediction of $\theta^*_k(t)$, i.e., the t^{th} estimation of k^{th} element of $\boldsymbol{\theta}^*$ base on $\mathbf{\Theta}^p_k(t)$.

Proposition 1. *Assume $[\epsilon^\theta_1, \ldots, \epsilon^\theta_m]^\top \sim \mathcal{N}(0, \sigma^\theta_k \mathbf{I}_m)$ and $\mathcal{F}^\theta_k$ is a GP with mean zero and covariance function $\mathcal{K}$, i.e., $\mathcal{F}^\theta_k \sim \mathcal{GP}(0, \mathcal{K}(\cdot,\cdot))$, then*

$$\theta^*_k(t) = \mathbf{\Sigma}_{\mathbf{x}^*\mathbf{X}^p}(\mathbf{\Sigma}_{\mathbf{X}^p\mathbf{X}^p} + \sigma^\theta_k \mathbf{I}_m)^{-1}\mathbf{\Theta}^p_k(t). \tag{14}$$

To find point and interval predictions for the new response y^*, we make T predictions based on the T samples we drew from the posterior (13) and the T predictions we made for the vector $\boldsymbol{\theta}^*$ using (14). Recall from Section 2 that $\mathcal{F}^s \sim \mathcal{GP}(0, \mathcal{K}(\cdot,\cdot))$; therefore, we can use the GP predictive distribution to derive the t^{th} prediction as

$$\begin{aligned}\mathcal{F}^s(\mathbf{x}^*, \boldsymbol{\theta}^*(t), \boldsymbol{\psi}^p(t)) \sim \mathcal{N}\Big(&\mathbf{\Sigma}_{\mathbf{v}^*(t)\mathbf{V}(t)}\big(\mathbf{\Sigma}_{\mathbf{V}(t)\mathbf{V}(t)} + \sigma^2(t)\mathbf{I}_m\big)^{-1}\mathbf{y}^p,\\ &\mathbf{\Sigma}_{\mathbf{v}^*(t)\mathbf{v}^*(t)} - \mathbf{\Sigma}_{\mathbf{v}^*(t)\mathbf{V}(t)}\big(\mathbf{\Sigma}_{\mathbf{V}(t)\mathbf{V}(t)} + \sigma^2(t)\mathbf{I}_m\big)^{-1}\mathbf{\Sigma}_{\mathbf{V}(t)\mathbf{v}^*(t)}\Big),\end{aligned} \tag{15}$$

where $\mathbf{v}^*(t) = [\mathbf{x}^{*\top}, \boldsymbol{\theta}^{*\top}(t), \boldsymbol{\psi}^{p\top}(t)]^\top$, $\mathbf{V}(t) = [\mathbf{X}^P, \mathbf{\Theta}^P(t), \mathbb{1}_{m\times d_\psi}\text{diag}(\boldsymbol{\psi}^p(t))]^\top$, and the covariance matrices are calculated using the t^{th} sample of the covariance parameters, namely $\boldsymbol{\ell}(t)$ and $\gamma(t)$.

Finally, we derive our prediction using distribution (15) as

$$\begin{aligned}\hat{\mu}^* &= \frac{1}{T}\sum_{t=1}^{T}\Big(\mathbf{\Sigma}_{\mathbf{v}^*(t)\mathbf{V}(t)}\big(\mathbf{\Sigma}_{\mathbf{V}(t)\mathbf{V}(t)} + \sigma^2(t)\mathbf{I}_m\big)^{-1}\mathbf{y}^p\Big),\\ \hat{\sigma}^{*2} &= \frac{1}{T^2}\sum_{t=1}^{T}\Big(\mathbf{\Sigma}_{\mathbf{v}^*(t)\mathbf{v}^*(t)} - \mathbf{\Sigma}_{\mathbf{v}^*(t)\mathbf{V}(t)}\big(\mathbf{\Sigma}_{\mathbf{V}(t)\mathbf{V}(t)} + \sigma^2(t)\mathbf{I}_m\big)^{-1}\mathbf{\Sigma}_{\mathbf{V}(t)\mathbf{v}^*(t)}\Big).\end{aligned}$$

6 Experimental results

In this section, we evaluate the performance of our methodology by testing it on three synthetic problems and two real problems. We use the root mean squared error (RMSE) as the measure of accuracy in prediction of responses and calibration vectors to compare the performance of the competing methodologies;

$$\text{RMSE}_y = \sqrt{\frac{1}{n^*}\sum_{q=1}^{n^*}(\hat{y}_q^* - y_q^*)^2} \quad \text{and} \quad \text{RMSE}_\theta = \sqrt{\frac{1}{n^*}\sum_{q=1}^{n^*}\left\|[\hat{\boldsymbol{\theta}}_q^{*^\top}, \hat{\boldsymbol{\psi}}_q^{*^\top}]^\top - [\bar{\boldsymbol{\theta}}_q^{*^\top}, \bar{\boldsymbol{\psi}}_q^{*^\top}]^\top\right\|_2^2},$$

where $\hat{y_q}^*$, $\hat{\boldsymbol{\theta}}_q^*$, and $\hat{\boldsymbol{\psi}}_q^*$ are the predicted response and calibration parameter values for the q^{th} test control variable.

Both the MCF and DCSUB formulations find anchor points for each dataset in less than two minutes on all the instances in our testbed, which shows that both formulations are fast in terms of computation time. To choose proper values of λ, r, and M for the GMST model, we use the following empirical approach. First, in order to have the same scale for the inputs and outputs in S and P, before constructing the calibration graph we standardize (divide by the range of each element) the control set $\{\mathbf{x}_i^p, \mathbf{x}_j^s \mid i \in \{1, 2, \ldots, m\}, j \in \{1, 2, \ldots, n\}\}$, the calibration set $\{\boldsymbol{\theta}_j^s, \boldsymbol{\psi}_j^s \mid j \in \{1, 2, \ldots, n\}\}$, and the response set $\{y_j^s, y_i^p \mid i \in \{1, 2, \ldots, m\}, j \in \{1, 2, \ldots, n\}\}$. We denote the standardized versions of $\mathbf{x}_i^p, \mathbf{x}_j^s, \boldsymbol{\theta}_j^s, \boldsymbol{\psi}_j^s, y_j^s$, and y_i^p by $\bar{\mathbf{x}}_i^p, \bar{\mathbf{x}}_j^s, \bar{\boldsymbol{\theta}}_j^s, \bar{\boldsymbol{\psi}}_j^s, \bar{y}_j^s$, and $\bar{y}_i^p$. Note that this standardization is only for finding the anchor points, and once the anchor points are chosen, we transform the data back to their original scales for Bayesian inference. After standardization, λ has to be less than two, otherwise the closeness of the calibration vectors is weighted as more important than closeness of the responses. We choose λ from $\{0.1, 0.5, 1, 1.5\}$ in our experiments. Moreover, for M to be a sufficiently large number, it should be larger than the sum of all the weights on the arcs that can be potentially in E_1, that is:

$$M \geq \sum_{j\in\{1,2,\ldots,n\}}\ \sum_{k\in\{1,2,\ldots,n\}}\ \sum_{i\in\{1,2,\ldots,m\}}\ \sum_{\ell\in\{1,2,\ldots,m\}} |\bar{y}_j^s - \bar{y}_i^p| + |\bar{y}_k^s - \bar{y}_\ell^p| + \lambda \left\|[\bar{\boldsymbol{\theta}}_j^{s^\top}, \bar{\boldsymbol{\psi}}_j^{s^\top}]^\top - [\bar{\boldsymbol{\theta}}_k^{s^\top}, \bar{\boldsymbol{\psi}}_k^{s^\top}]^\top\right\|_2.$$

Finally, to choose a proper value for r, we plot the pairwise Euclidean distances between all $\bar{\mathbf{x}}^p$ vectors. Then, we use the fact that each point on this plot represents an existing distance between the centers of two clusters in the calibration graph. Therefore, an upper bound on a group of smallest distances, which are close together and disjoint from other groups of distances, can be used as a proper value for r.

6.1 Description of the calibration problems

Table 1 describes synthetic problems used in this study to test the performance of our model on different settings of the calibration problem. The first synthetic problem has one functional calibration variable and one control variable. The second problem has an additional control variable, and the third problem has an additional global calibration variable. We also evaluate the performance of our model on a high dimensional synthetic problem (that is when d^x, the dimension of the domain of the calibration variables, is large), the results of which are presented in Appendix D. We note that since in practical settings physical observations are noise contaminated, we add a noise drawn from $\mathcal{N}(0, 0.05)$ to each generated y^p.

Problem	$\mathcal{F}^s(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\psi})$	$\mathcal{F}^p(\mathbf{x})$	$\mathcal{F}^\theta(\mathbf{x})$	$\boldsymbol{\psi}$
1	$\theta \exp(-0.05x^2)(\sin(x)^2 + 1)$	$\exp(-0.05x^2 - 0.05x) * (\sin(x)^2 + 1)$	$\exp(-0.05x)$	-
2	$0.4(x_1^2 + x_2^2)\sin^2(0.7x_2)\frac{x_1+x_2}{\theta^2+1}$	$0.4(x_1^2 + x_2^2)\sin^2(0.7x_2)$	$(x_1 + x_2 - 1)^{0.5}$	-
3	$\theta + \psi x^2$	$2\sqrt{x} + 2.5x^2$	$2\sqrt{x}$	2.5

Table 1: Calibration functions defined for the three synthetic problems

For the first synthetic problem, we locate $m = 6$ control vectors, $\mathbf{x}^p$, at locations $\{0.5, 1.5, 2.5, 3.5, 4.5, 5.5\}$. Then for each $\mathbf{x}^p$, we randomly sample five functional calibration vectors from the interval $[0, 2]$; therefore, we have a total of $n = 30$ computational data points. Finally, we sample 12 random test control vectors, $\mathbf{x}^*$, from the line segment $[0, 6]$ to form a test dataset.

For the second synthetic problem, we locate $m = 16$ control vectors, $\mathbf{x}^p$, uniformly on the square $[0, 3.5] \times [0, 3.5]$. Then for each $\mathbf{x}^p$, we sample 10 functional calibration vectors randomly from the interval $[0, 5]$; therefore, we have a total of $n = 160$ computational data points. Finally, we sample 10 random test control vectors, $\mathbf{x}^*$, from the same square $[0, 3.5] \times [0, 3.5]$ to form a test dataset.

For the third synthetic problem, we follow the setting used by Brown and Atamturktur (2018), where we choose $m = 15$ control vectors for training at locations $\{0, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95\}$, and use five physical control vectors for testing at locations $\{0.45, 0.50, 0.55, 0.60, 0.65\}$. We sample 10 functional calibration vectors for each $\mathbf{x}^p$ from the square $[0, 5] \times [0, 5]$; therefore, we have a total of $n = 150$ computational data points.

The first real problem from a spot welding application was originally introduced by Bayarri et al. (2007). This problem has three control variables and one calibration variable. The dataset associated with spot welding contains 12 and 35 data points sampled from the physical experiments and the computation model, respectively. The second real problem studied by Pourhabib et al. (2015) has one control variable and one calibration variable, and the associated dataset contains 11 and 150 data points sampled from the physical experiments and the computation model, respectively. This instance arises from a PVA-treated buckypaper fabrication process.

For the real problems, we partition the sets of physical data points using four-fold cross validation to form training and test datasets. Therefore, for each iteration of cross validation for the spot welding dataset, we have eight physical data points in the training set and four physical data points in the test set. Similarly, for the PVA dataset we have eight to nine physical data points in the training set, and two to three data points in the test set in each iteration of cross validation. We note that the cross validation does not affect the size of the computational datasets, i.e., $n = 35$ for spot welding and $n = 150$ for the PVA dataset.

6.2 Results

We compare the results of our proposed methods with competing functional calibration methodologies. These include non-parametric functional calibration (NFC) (Pourhabib et al., 2018), parametric functional calibration (PFC) (Pourhabib et al., 2015), and non-parametric Bayesian calibration (NBC) (Brown and Atamturktur, 2018), which all require surrogate modeling for handling expensive computational models. When we use the generalized minimum spanning tree model on a calibration digraph to find a set of anchor points, we refer to our approach as BMNC. If we use the shortest path model on a calibration digraph, we call the approach BMNC-DAG.

For each of the aforementioned problems, we choose the values of λ, r, and M following the empirical approach explained in Section 6 (See Table 2). Tables 3 and 4 compare the performance of BNMC and BNMC-DAG in terms of RMSE_θ, RMSE_y, and computation time with the other competing methodologies.

For the first synthetic problem, the second and third columns of Table 3 show that BNMC and BNMC-DAG both perform more accurately than the other methodologies in terms of RMSE_y, and they have the same order of accuracy in terms of RMSE_θ. Moreover, Table 4 shows that BNMC-DAG performs slightly faster than BNMC. However, because we only have $n = 30$ computational and $m = 6$ physical data points, this difference is not very large. To better compare the computational costs of the approach using the shortest path and the GMST models, we choose values of ns from the set $\{100, 200, 300, 400, 500\}$ and run both BNMC and BNMC-DAG. As expected, the shortest path model in BNMC-DAG finds the anchor points for all the values of n in less than a second, whereas GMST requires $3.6, 21.2, 65.3, 144.7,$ and 323.5 seconds for $n \in \{100, 200, 300, 400, 500\}$, respectively.

For the second synthetic problem, because the dimension of $\mathbf{x}^p$ is greater than one, we cannot apply BNMC-DAG. However, the fourth and fifth columns of Table 3 show that BNMC outperforms the other methodologies in terms of RMSE_y and has the second best accuracy in terms of RMSE_θ.

For the third synthetic problem, we only compare the results of NBC and BNMC, since the codes for NFC and PFC are written only for univariate calibration problems. The sixth and the seventh columns of Table 3 show that BNMC outperforms NBC both in terms of RMSE_y and RMSE_θ. We note that the reported RMSE_y for NBC in (Brown and Atamturktur, 2018) under the cheap computational code assumption is 0.0538, which is a better accuracy compared to that of BNMC; however, here BNMC is superior when NBC uses surrogate modeling.

Since the true values of the calibration parameters are unknown for the real problems, we compare the results only in terms of RMSE_y. The eighth column (PVA) of Table 3 shows that BNMC, BNMC-DAG, and NBC have the same order of accuracy and perform better than NFC and PFC. Finally, we observe in the last column of Table 3 that for the Spot Welding problem BNMC outperforms the other competing methodologies with a large margin. We attribute this performance to the capability of BNMC in handling expensive computational models with a small number of computational data points.

Parameter	1st synthetic problem	2nd synthetic problem	3rd synthetic problem	PVA	Spot Welding
λ	0.5	0.5	0.5	0.5	0.5
r	1.5	0.9	0.16	0.5	4
M	10^5	10^5	10^5	10^5	10^5

Table 2: The calibration graph parameters for the five calibration problems.

Methodology	1st synthetic problem		2nd synthetic problem		3rd synthetic problem		PVA	Spot Welding
	RMSE_y	RMSE_θ	RMSE_y	RMSE_θ	RMSE_y	RMSE_θ	RMSE_y	RMSE_y
NFC	0.184	0.254	0.143	0.202	-	-	0.379	0.683
PFC	0.226	0.244	0.296	0.390	-	-	0.450	1.115
NBC	0.162	0.356	0.132	0.627	0.172	0.426	0.281	0.516
BNMC	0.098	0.271	0.076	0.356	0.063	0.354	0.296	0.409
BNMC-DAG	0.098	0.265	-	-	-	-	0.288	-

Table 3: RMSE_θ and RMSE_y of different methodologies for the five calibration problems.

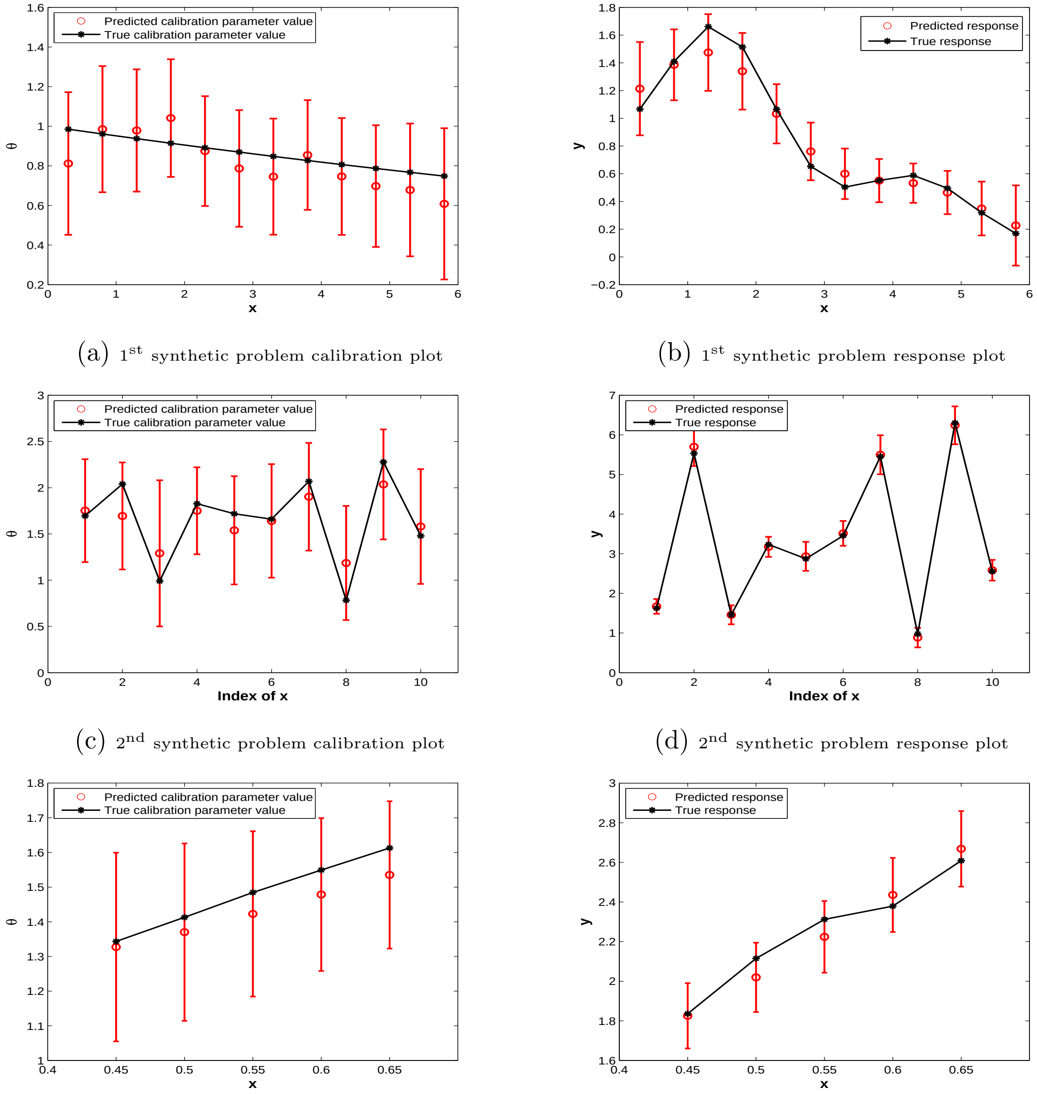
Methodology	1st synthetic problem	2nd synthetic problem	3rd synthetic problem	PVA	Spot Welding
NFC	2.88	8.02	-	3.29	0.77
PFC	18.04	253.08	-	176.95	19.06
NBC	45.26	307.31	230.19	180.92	68.04
BNMC	125.15	169.23	147.01	159.84	117.53
BNMC-DAG	123.91	-	-	144.54	-

Table 4: The computation times (in seconds) for the five calibration problems.

Figure 4 shows the 95% confidence interval predictions for the responses and the functional calibration parameter for the test datasets of the synthetic problems. Note that since $\mathbf{x}^p \in \mathbb{R}^2$ for the second synthetic problem, we plot the predicted values against their indices in Figures 4c and 4d, and connect the data points to each other for better visualization. Moreover, although we added white noise to response variables to mimic real world processes, we show the denoised responses for clearer illustration.

As noted in Section 2, due to the limited number of samples we collect from the computational model, we cannot accurately recover $\mathcal{F}^\theta$, but the way we train the hyper-parameters of the GP aims to compensate for this limitation. We can observe this in Figure 4, where the predictions of the response values have better accuracy and tighter confidence intervals compared to those of functional calibration parameter values.

For illustration, Figure 5 shows the 95% confidence interval predictions for one of the test datasets created in the cross validation process for each of the spot welding and the PVA problems. Since we do not know the true functional calibration parameter values, we cannot provide a similar plot for the calibration predictions. Similar to Figures 4c and 4d, we plot the predicted values against their indices in Figure 5a for better visualization.



(a) 1^{st} synthetic problem calibration plot

(b) 1^{st} synthetic problem response plot

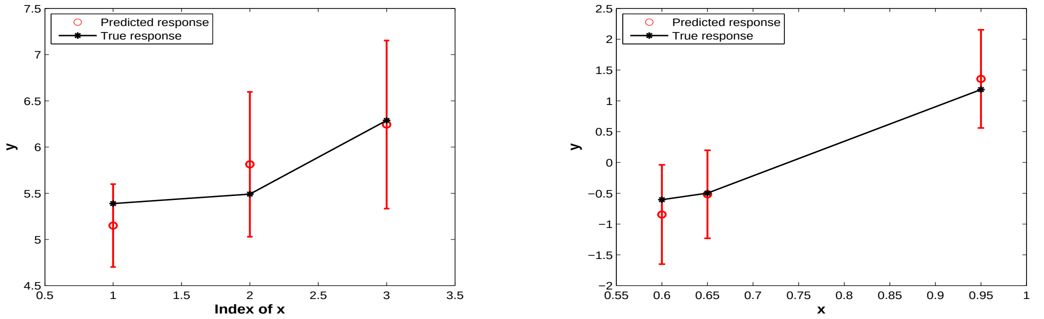
(c) 2^{nd} synthetic problem calibration plot

(d) 2^{nd} synthetic problem response plot

(e) 3^{rd} synthetic problem calibration plot

(f) 3^{rd} synthetic problem response plot

Figure 4: The 95% confidence interval predictions for functional calibration parameter values and responses for the test datasets of the synthetic problems.



(a) Spot welding response plot

(b) PVA response plot

Figure 5: The 95% confidence interval predictions for one of the test datasets created in the cross validation process for each of the spot welding and the PVA problems.

We refer the reader to Appendix E for an analysis of the residuals to validate the assumptions made in our proposed approach, especially those made in equations (2) and (3).

7 Concluding remarks

We proposed a Bayesian non-isometric matching calibration model for expensive computational models. A limited budget to evaluate computational models led to the use of a GP, which was trained *during* the calibration procedure. We used Bayesian statistics to simultaneously train the hyper-parameters of the GP's covariance function and make inferences on the calibration parameters associated with the physical data points. To construct informative prior distributions for our new approach, we used a geometric interpretation of calibration based on non-isometric curve to surface matching. This point of view enabled us to develop graph-theoretic approaches to address the problem of finding a set of anchor points used in constructing informative prior distributions. For the special case of a single control and calibration variable, we introduced a shortest path model on a directed acyclic calibration graph to tackle the problem of finding anchor points, while for the general case, we introduced the generalized minimum spanning tree model. Our numerical experiments conducted on four benchmark calibration problems showed that our approach outperformed the existing calibration models under the assumption of expensive computational models.

The framework developed in this paper could be extended in several ways. We only considered a single computational data point to construct a prior distribution for each calibration parameter;

however, information of multiple computational data points could be taken into account. An implementation of this idea, of course, requires developing new combinatorial optimization techniques capable of choosing an appropriate number of computational data points. Another interesting research path would be to consider data uncertainty formally in the calibration graph model instead of using a deterministic calibration graph model, and then using Bayesian inference to deal with the uncertainty in the data. Moreover, one may consider including an independent discrepancy function in equation (3), as noted in Section 2. Finally, the proposed approach would potentially benefit from using cross-validation for the selection of the tuning parameters such as λ, r, and M.

Acknowledgment

Austin Buchanan is supported by the National Science Foundation under Grant No. 1662757. This work was completed utilizing the High Performance Computing Center facilities of Oklahoma State University at Stillwater. The authors also acknowledge Chuck Zhang from Georgia Institute of Technology for providing the authors with the PVA-treated buckypaper fabrication data.

References

Baltsavias, E., A. Gruen, H. Eisenbeiss, L. Zhang, and L. Waser (2008). High-quality image matching and automated generation of 3D tree models. *International Journal of Remote Sensing 29*(5), 1243–1259.

Bayarri, M. J., J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin, and J. Tu (2007). A framework for validation of computer models. *Technometrics 49*(2), 138–154.

Bellman, R. (1958). On a routing problem. *Quarterly of Applied Mathematics 16*(1), 87–90.

Bertsimas, D. and R. Weismantel (2005). *Optimization Over Integers*. Belmont, Massachusetts: Dynamic Ideas.

Bronstein, A. M., M. M. Bronstein, and R. Kimmel (2003). Expression-invariant 3d face recognition. In *International Conference on Audio-and Video-based Biometric Person Authentication*, pp. 62–70. Springer.

Bronstein, A. M., M. M. Bronstein, and R. Kimmel (2005). Three-dimensional face recognition. *International Journal of Computer Vision 64*(1), 5–30.

Brown, D. A. and S. Atamturktur (2018). Nonparametric functional calibration of computer models. *Statistica Sinica 28*, 721–742.

Buchanan, A., J. S. Sung, S. Butenko, and E. L. Pasiliao (2015). An integer programming approach for fault-tolerant connected dominating sets. *INFORMS Journal on Computing 27*(1), 178–188.

Craig, P. S., M. Goldstein, J. C. Rougier, and A. H. Seheult (2001). Bayesian forecasting for complex systems using computer simulators. *Journal of the American Statistical Association 96*(454), 717–729.

Ezzat, A. A., A. Pourhabib, and Y. Ding (2018). Sequential design for functional calibration of computer models. *Technometrics 60*(3), 286–296.

Fang, K.-T., R. Li, and A. Sudjianto (2005). *Design and modeling for computer experiments.* CRC Press.

Feremans, C., M. Labbé, and G. Laporte (2002). A comparative analysis of several formulations for the generalized minimum spanning tree problem. *Networks 39*(1), 29–34.

Gelfand, A. E., S. E. Hills, A. Racine-Poon, and A. F. Smith (1990). Illustration of Bayesian inference in normal data models using Gibbs sampling. *Journal of the American Statistical Association 85*(412), 972–985.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin (2014). *Bayesian Data Analysis*, Volume 2. Chapman & Hall/CRC Boca Raton, FL, USA.

Goldstein, M. and J. Rougier (2009). Reified Bayesian modelling and inference for physical systems. *Journal of Statistical Planning and Inference 139*(3), 1221–1239.

Gruen, A. and D. Akca (2005). Least squares 3D surface and curve matching. *ISPRS Journal of Photogrammetry and Remote Sensing 59*(3), 151–174.

Han, G., T. J. Santner, and J. J. Rawlinson (2009). Simultaneous determination of tuning and calibration parameters for computer experiments. *Technometrics 51*(4), 464–474.

Higdon, D., J. Gattiker, B. Williams, and M. Rightley (2008). Computer model calibration using high-dimensional output. *Journal of the American Statistical Association 103*(482), 570–583.

Higdon, D., M. Kennedy, J. C. Cavendish, J. A. Cafeo, and R. D. Ryne (2004). Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing 26*(2), 448–466.

Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences 186*(1007), 453–461.

Joseph, V. R. and S. N. Melkote (2009). Statistical adjustments to engineering models. *Journal of Quality Technology 41*(4), 362.

Kennedy, M. C. and A. O'Hagan (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63*(3), 425–464.

Lawler, E. (1976). *Combinatorial Optimization: Networks and Matroids.* New York: Holt, Rinehart, and Winston.

Ljung, G. M. and G. E. Box (1978). On a measure of lack of fit in time series models. *Biometrika 65*(2), 297–303.

Loeppky, J., D. Bingham, and W. Welch (2006). Computer model calibration or tuning in practice. Technical report, University of British Columbia, Vancouver, BC, CA.

Lu, Y., E. Moradi, and B. Balasundaram (2018, November). Correction to: Finding a maximum k-club using the k-clique formulation and canonical hypercube cuts. *Optimization Letters 12*(8), 1959–1969.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics 21*(6), 1087–1092.

Moradi, E. and B. Balasundaram (2018, November). Finding a maximum k-club using the k-clique formulation and canonical hypercube cuts. *Optimization Letters 12*(8), 1947–1957.

Myung, Y.-S., C.-H. Lee, and D.-W. Tcha (1995). On the generalized minimum spanning tree problem. *Networks 26*(4), 231–241.

Plumlee, M. and V. R. Joseph (2018). Orthogonal Gaussian process models. *Statistica Sinica 28*(2), 601–619.

Plumlee, M., V. R. Joseph, and H. Yang (2016). Calibrating functional parameters in the ion channel models of cardiac cells. *Journal of the American Statistical Association 111*(514), 500–509.

Pop, P. C. (2004). New models of the generalized minimum spanning tree problem. *Journal of Mathematical Modelling and Algorithms 3*(2), 153–166.

Pop, P. C., W. Kern, and G. Still (2006). A new relaxation method for the generalized minimum spanning tree problem. *European Journal of Operational Research 170*(3), 900–908.

Pourhabib, A., J. Z. Huang, K. Wang, C. Zhang, B. Wang, and Y. Ding (2015). Modulus prediction of buckypaper based on multi-fidelity analysis involving latent variables. *IIE Transactions 47*(2), 141–152.

Pourhabib, A., R. Tuo, S. He, J. Z. Huang, and Y. Ding (2018). Functional calibration of computer models. Manuscript.

Pratola, M. T., S. R. Sain, D. Bingham, M. Wiltberger, and E. J. Rigler (2013). Fast sequential computer model calibration of large nonstationary spatial-temporal processes. *Technometrics 55*(2), 232–242.

Rasmussen, C. E. (2004). Gaussian Processes for Machine Learning. In *Advanced Lectures on Machine Learning*, pp. 63–71. Springer.

Reese, C. S., A. G. Wilson, M. Hamada, H. F. Martz, and K. J. Ryan (2004). Integrated analysis of computer and physical experiments. *Technometrics 46*(2), 153–164.

Santner, T. J., B. J. Williams, and W. I. Notz (2013). *The Design and Analysis of Computer Experiments.* Springer Science & Business Media.

Schölkopf, B., R. Herbrich, and A. J. Smola (2001). A generalized representer theorem. In *International Conference on Computational Learning Theory*, pp. 416–426. Springer.

Shapiro, S. S. and M. B. Wilk (1965). An analysis of variance test for normality (complete samples). *Biometrika 52*(3/4), 591–611.

Tuo, R. and C. F. J. Wu (2015). Efficient calibration for imperfect computer models. *The Annals of Statistics 43*(6), 2331–2352.

Tuo, R. and C. F. J. Wu (2016). A theoretical framework for calibration in computer models: Parametrization, estimation and convergence properties. *SIAM/ASA Journal on Uncertainty Quantification 4*(1), 767–795.

Williams, B., D. Higdon, J. Gattiker, L. Moore, M. McKay, S. Keller-McNulty, et al. (2006). Combining experimental data and computer simulations, with an application to flyer plate experiments. *Bayesian Analysis 1*(4), 765–792.

Xiong, Y., W. Chen, K.-L. Tsui, and D. W. Apley (2009). A better understanding of model updating strategies in validating engineering models. *Computer Methods in Applied Mechanics and Engineering 198*(15), 1327–1337.

Appendix A Full and conditional posterior distributions

First, we explain the prior distributions for parameters in (13). We have

$$\begin{aligned}
&\nu_k^2 \sim \frac{1}{\nu_k^2}, && \forall k \in \{1, \dots, d^\theta\},\\
&\tau_h^2 \sim \text{Inv-Gamma}(\alpha_\tau, \beta_\tau), && \forall h \in \{1, \dots, d^\psi\},\\
&\ell_j \sim \text{Log-Gamma}(\alpha_\ell, \beta_\ell), && \forall j \in \{1, \dots, d^x + d^\theta + d^\psi\},\\
&\gamma \sim \text{Log-Uniform},\\
&\sigma^2 \sim \text{Log-Uniform}.
\end{aligned}$$

For each ν_k^2 we use a flat Jeffreys prior (Jeffreys, 1946), which is an inverse gamma distribution with zero value for both the shape and the scale parameter. For each τ_h^2 we choose a weak inverse gamma distribution, i.e., an inverse gamma with large variance, with $\alpha_\tau = 2.1$ and $\beta_\tau = 10$ as its parameters. Note that both of these prior distributions are conjugate for their associated parameters in the posterior distribution. Moreover, as recommended by Gelman et al. (2014), to improve the identifiability of the model we use the prior distributions on the logarithmic scale for parameters of the GP part of the model. Therefore, for σ^2 and γ we use a flat log-uniform distribution and for each ℓ_j we use a log-gamma distribution with the parameters $\alpha_\ell = \beta_\ell = 2$.

Using the prior distributions explained above the posterior (13) can be written as

$$\begin{aligned}
&\pi(\boldsymbol{\Theta}^p, \boldsymbol{\psi}^p, \boldsymbol{\nu}^2, \boldsymbol{\tau}^2, \boldsymbol{\ell}, \gamma, \sigma^2 \mid \mathbf{y}^p, \mathbf{X}^p, \boldsymbol{\Theta}^a, \boldsymbol{\psi}^a)\\
&\propto |\boldsymbol{\Sigma} + \sigma \mathbf{I}_m|^{-0.5} \exp\left\{\frac{-1}{2} \mathbf{y}^{p\top} (\boldsymbol{\Sigma} + \sigma \mathbf{I}_m)^{-1} \mathbf{y}^p\right\}\\
&\times \prod_k (\nu_k^2)^{-m/2-1} \exp\left\{\frac{-1}{2\nu_k^2} (\boldsymbol{\Theta}_k^p - \boldsymbol{\Theta}_k^a)^\top (\boldsymbol{\Theta}_k^p - \boldsymbol{\Theta}_k^a)\right\}\\
&\times \prod_h (\tau_h^2)^{-1/2-\alpha_\tau-1} \exp\left\{\frac{-1}{2\tau_h^2} (\psi_h^p - \psi_h^a)^2\right\} \exp\left\{\frac{-\beta_\tau}{\tau_h^2}\right\}\\
&\times \prod_j \frac{1}{\ell_j} \log(\ell_j)^{\alpha_\ell - 1} \exp\{-\beta_\ell \log(\ell_j)\}\\
&\times \frac{1}{\gamma} \times \frac{1}{\sigma^2}.
\end{aligned} \tag{16}$$

We use Gibbs sampling (Gelfand et al., 1990) to sequentially sample from the full conditional posterior distributions. Here we present the full conditional distribution for each of the parameters in (16):

$$
\begin{aligned}
\pi(\Theta^p_{ki} \mid \cdot) &\propto |\mathbf{\Sigma} + \sigma \mathbf{I}_m|^{-0.5} \\
&\qquad \times \exp\left\{ \frac{-1}{2}\Big(\mathbf{y}^{p\top}(\mathbf{\Sigma} + \sigma \mathbf{I}_m)^{-1}\mathbf{y}^p + \frac{1}{\nu_k^2}(\Theta^p_{ki} - \Theta^a_{ki})^2 \Big) \right\} \quad \forall k \in \{1, \dots, d^\theta\}, \forall i \in \{1, 2, \dots, m\} \\
\pi(\psi^p_h \mid \cdot) &\propto |\mathbf{\Sigma} + \sigma \mathbf{I}_m|^{-0.5} \\
&\qquad \times \exp\left\{ \frac{-1}{2}\Big(\mathbf{y}^{p\top}(\mathbf{\Sigma} + \sigma \mathbf{I}_m)^{-1}\mathbf{y}^p + \frac{1}{\tau_k^2}(\psi^p_h - \psi^a_h)^2 \Big) \right\} \qquad \forall h \in \{1, \dots, d^\psi\} \\
\pi(\nu_k^2 \mid \cdot) &\propto (\nu_k^2)^{-m/2-1} \exp\left\{ \frac{-1}{2\nu_k^2}(\mathbf{\Theta}^p_k - \mathbf{\Theta}^a_k)^\top (\mathbf{\Theta}^p_k - \mathbf{\Theta}^a_k) \right\} \qquad \forall k \in \{1, \dots, d^\theta\} \\
\pi(\tau_h^2 \mid \cdot) &\propto (\tau_h^2)^{-1/2-\alpha_\tau-1} \exp\left\{ \frac{-1}{2\tau_h^2}\Big((\psi^p_h - \psi^a_h)^2 + 2\beta_\tau \Big) \right\} \qquad \forall h \in \{1, \dots, d^\psi\} \\
\pi(\ell_j \mid \cdot) &\propto \frac{\log(\ell_j)^{\alpha_\ell - 1} |\mathbf{\Sigma} + \sigma \mathbf{I}_m|^{-0.5}}{\ell_j} \\
&\qquad \times \exp\left\{ \frac{-1}{2}\Big(\mathbf{y}^{p\top}(\mathbf{\Sigma} + \sigma \mathbf{I}_m)^{-1}\mathbf{y}^p + 2\beta_\ell \log(\ell_j) \Big) \right\} \qquad \forall j \in \{1, \dots, d^x + d^\theta + d^\psi\} \\
\pi(\gamma \mid \cdot) &\propto \frac{|\mathbf{\Sigma} + \sigma \mathbf{I}_m|^{-0.5}}{\gamma} \exp\left\{ \frac{-1}{2} \mathbf{y}^{p\top}(\mathbf{\Sigma} + \sigma \mathbf{I}_m)^{-1}\mathbf{y}^p \right\} \\
\pi(\sigma^2 \mid \cdot) &\propto \frac{|\mathbf{\Sigma} + \sigma \mathbf{I}_m|^{-0.5}}{\sigma^2} \exp\left\{ \frac{-1}{2} \mathbf{y}^{p\top}(\mathbf{\Sigma} + \sigma \mathbf{I}_m)^{-1}\mathbf{y}^p \right\},
\end{aligned}
$$

where the notation $(\mid \cdot)$ denotes the conditioning over every other parameters of posterior (16).

We note that $\nu_k^2 \mid \cdot$ and $\tau_k^2 \mid \cdot$ have inverse gamma distributions with parameters $\left(\frac{m}{2}, \frac{(\mathbf{\Theta}^p_k - \mathbf{\Theta}^a_k)^\top (\mathbf{\Theta}^p_k - \mathbf{\Theta}^a_k)}{2}\right)$ and $\left(\frac{1}{2} + \alpha_\tau, \frac{(\psi^p_h - \psi^a_h)^2 + 2\beta_\tau}{2}\right)$, respectively. However, the rest of the conditional distributions do not have closed form distributions; therefore, we take Metropolis-Hastings (Metropolis et al., 1953) steps to sample from these distributions during the sampling process.

Appendix B Proof of Lemma 1 and Proposition 1

Proof of Lemma 1. It suffices to show that if (u, v) is an edge, then $u < v$, which is trivially true when $u = 0$ or $v = n + 1$. For distinct vertices $u, v \in V^0$, note that $u < v$ if and only if $\mathbf{x}^s_u < \mathbf{x}^s_v$ as we have assumed the points in S to be strictly ordered. Suppose, $u \in C_i$ and $v \in C_{i+1}$ for some $i \in \{1, 2, \dots, m-1\}$. Hence, $\mathbf{x}^p_i < \mathbf{x}^p_{i+1}$, and from equation (7) we can conclude that the distinct points $\mathbf{x}^s_u$ and $\mathbf{x}^s_v$ satisfy $\mathbf{x}^s_u \le \frac{1}{2}(\mathbf{x}^p_i + \mathbf{x}^p_{i+1}) \le \mathbf{x}^s_v$. □

Proof of Proposition 1. To obtain the distribution of $\mathbf{\Theta}_k^p(t)$, we note that the assumptions $[\epsilon_1^\theta, \ldots, \epsilon_m^\theta]^\top \sim \mathcal{N}(0, \sigma_k^\theta \mathbf{I}_m)$ and $\mathcal{F}_k^\theta \sim \mathcal{GP}(0, \mathcal{K}(\cdot,\cdot))$ imply that

$$\mathbf{\Theta}_k^p(t) \sim \mathcal{N}(0, \mathbf{\Sigma}_{\mathbf{X}^p\mathbf{X}^p} + \sigma_k^\theta \mathbf{I}_m), \tag{17}$$

where we denote the covariance between columns of matrices $\mathbf{Z}$ and $\mathbf{Z}'$ by $\mathbf{\Sigma}_{\mathbf{ZZ}'}$. We add the variance of error, σ_k^θ, to preserve the smoothness of $\mathcal{F}_k^\theta$; otherwise, our approach would obtain $\mathcal{F}_k^\theta$ as the interpolation of the elements of $\mathbf{\Theta}_k^p(t)$.

Further, by the GP assumption on $\mathcal{F}_k^\theta$, we obtain the joint distribution of $\mathbf{\Theta}_k^p(t)$ and the prediction of θ_k^* for the t^{th} draw, which we denote by $\theta_k^*(t)$, as

$$\begin{bmatrix} \mathbf{\Theta}_k^p(t) \\ \theta_k^*(t) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{X}^p\mathbf{X}^p} + \sigma_k^\theta \mathbf{I}_m & \mathbf{\Sigma}_{\mathbf{X}^p\mathbf{x}^*} \\ \mathbf{\Sigma}_{\mathbf{x}^*\mathbf{X}^p} & \mathbf{\Sigma}_{\mathbf{x}^*\mathbf{x}^*} \end{bmatrix}\right). \tag{18}$$

By conditioning on $\mathbf{\Theta}_k^p(t)$ in (18), the point prediction of $\theta_k^*(t)$ is obtained as

$$\theta_k^*(t) = \mathbf{\Sigma}_{\mathbf{x}^*\mathbf{X}^p}(\mathbf{\Sigma}_{\mathbf{X}^p\mathbf{X}^p} + \sigma_k^\theta \mathbf{I}_m)^{-1}\mathbf{\Theta}_k^p(t).$$

□

We note that for each prediction in (14), the hyper-parameters of the covariance function used in (17) should be tuned, which can be achieved by maximizing the logarithm of likelihood corresponding to (17) (Rasmussen, 2004):

$$\log(\pi(\Theta_k^p(t))) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{\Sigma}_{\mathbf{X}^p\mathbf{X}^p} + \sigma_k^\theta \mathbf{I}_m| - \frac{1}{2}\mathbf{\Theta}_k^{p^\top}(t)(\mathbf{\Sigma}_{\mathbf{X}^p\mathbf{X}^p} + \sigma_k^\theta \mathbf{I}_m)^{-1}\mathbf{\Theta}_k^p(t).$$

Appendix C Integer programming formulations

The GMST problem can also be viewed as a special case of the *generalized minimum spanning arborescence* (GMSA) problem, which is defined on a directed graph with its vertex set partitioned into clusters. Here we seek an arborescence (i.e., directed out-tree) of minimum weight, rooted at some vertex in a specified cluster that contains exactly one vertex per cluster. We can transform the GMST problem on graph $G = (V, E)$ to the GMSA problem by replacing each undirected edge $\{i, j\} \in E$ with directed anti-parallel arcs (i, j) and (j, i). Then, each arc is assigned the same weight as the corresponding undirected edge, and we arbitrarily choose one of the clusters to contain the root.

For the remainder of this discussion, we use the directed graph $\overset{\leftrightarrow}{G} = (V, A)$ corresponding to the calibration graph $G = (V, E)$, where $A \coloneqq \bigcup_{\{u,v\} \in E} \{(u, v), (v, u)\}$. The edge weight of each $e = \{u, v\} \in E$ is duplicated as arc-weights $w_{uv} = w_{vu} \coloneqq w_e$. Recall that the vertex set V is partitioned into clusters, say $C_1, \ldots, C_m$. We require the arborescence to be rooted at some vertex in C_1. We use binary decision vectors $q \in \{0, 1\}^{|A|}$ and $b \in \{0, 1\}^{|V|}$ to denote the incidence vectors of the arcs and vertices included in the arborescence, respectively.

$$\textbf{(DCSUB)} \quad \min \sum_{(u,v) \in A} w_{uv} q_{uv} \tag{19a}$$

$$\text{subject to:} \quad \sum_{v \in C_i} b_v = 1 \quad \forall i \in \{1, 2, \ldots, m\} \tag{19b}$$

$$q_{uv} + q_{vu} \leq 1 \quad \forall (u, v), (v, u) \in A \tag{19c}$$

$$\sum_{(u,v) \in A} q_{uv} = m - 1 \tag{19d}$$

$$\sum_{(u,v) \in A : u,v \in Q} q_{uv} \leq \sum_{v \in Q} b_v - 1 \quad \forall Q \subset V \mid Q \supset C_i \text{ for some } i \in \{1, 2, \ldots, m\} \tag{19e}$$

$$\sum_{u:(u,v) \in A} q_{uv} = b_v \quad \forall v \in V \backslash C_1 \tag{19f}$$

$$b \in \{0, 1\}^{|V|}, q \in \{0, 1\}^{|A|}. \tag{19g}$$

Constraints (19b) enforce that the model chooses exactly one vertex from each cluster, constraints (19d) ensure that exactly $m - 1$ arcs from A are selected, and constraints (19c) ensure that these correspond to $m - 1$ distinct edges in E. Cluster subpacking constraints (19e) prevent solutions that contain cycles and were shown by Feremans et al. (2002) to dominate the subtour elimination constraints used by Myung et al. (1995):

$$\sum_{(v,w) \in A : v,w \in S} q_{vw} \leq \sum_{v \in S \backslash \{u\}} b_v \quad \forall u \in S \subset V,\ 2 \leq |S| \leq |V| - 1.$$

Finally, constraints (19f) ensure that every non-root vertex selected by the solution has exactly one incoming edge and every vertex outside C_1 that is not selected will have no incoming arcs. Combined with the requirement that we choose exactly m vertices and $m-1$ arcs without creating cycles, this ensures that we obtain an arborescence rooted at some vertex inside C_1. As discussed in Section 4.1, a computationally effective approach for solving the GMST problem using this formulation requires delayed constraint generation. This approach starts by relaxing formulation (19) by omitting constraints (19e), and adding them "on the fly" during the progress of the branch-and-bound algorithm.

Next we present the multi-commodity flow (MCF) formulation for the GMSA problem that avoids using exponentially many constraints, but uses an additional set of variables (Myung et al., 1995). The MCF formulation treats every vertex $v \in C_1$ to have supply b_v for each commodity $i \in \{2, \ldots, m\}$ corresponding to the remaining clusters; it treats every $v \in C_i$ to have a demand of b_v for commodity i. Suppose $b_v = 1$ for some $v \in C_i$, then a path must be traced from the root selected in C_1 to deliver commodity i. We use the additional set of commodity-flow variables f^i_{uv} to denote the amount of commodity $i \in \{2, 3, \ldots, m\}$ flowing on arc $(u, v) \in A$.

$$\textbf{(MCF)} \quad \min \sum_{(u,v)\in A} w_{uv} q_{uv} \tag{20a}$$

subject to: $\quad$ (19b), (19c), (19d), (19g)

$$\sum_{w:(v,w)\in A} f^i_{vw} - \sum_{u:(u,v)\in A} f^i_{uv} = \left\{ \begin{array}{cc} b_v, & \forall v \in C_1 \\ -b_v, & \forall v \in C_i \\ 0, & \forall v \notin C_1 \cup C_i \end{array} \right\} \quad \forall i \in \{2, \ldots, m\} \tag{20b}$$

$$0 \le f^i_{uv} \le q_{uv} \qquad \forall (u,v) \in A, i \in \{2, \ldots, m\}. \tag{20c}$$

Constraints (20b) are flow-balance constraints for each commodity, and constraints (20c) prevent flows on the edges that are not selected.

Formulations (19) and (20) are both equally good in terms of the tightness of the LP relaxations as the projection of the LP relaxation of the latter onto the (b, q)-space is the same as the LP relaxation of the former (Feremans et al., 2002).

Appendix D A high dimensional case study

In this section, we evaluate the performance of the competing calibration methodologies on a 10 dimensional problem, that is when the dimension of the domain of the control variables, $d^x = 10$. For this problem, we assume that the computational model has the form

$$\mathcal{F}^s(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\psi}) = \mathcal{G}(\mathbf{x}) + \boldsymbol{\theta},$$

where $\mathcal{G} : [0,1]^{10} \longrightarrow \mathbb{R}$ is a Gaussian process with a zero mean and covariance function (4) with parameters $\gamma = 1$ and $\ell^T = [11.1, 6.2, 4, 2.7, 2, 1.5, 1.2, 1, 0.8, 0.6]$. To build $\mathcal{G}$, we first generate 5,000 samples from the cube $[0,1]^{10}$ and form the matrix $\mathbf{X}$ whose size is 5000×10. We then generate 1000 samples from $\mathcal{N}(0,1)$ and form the vector $\mathbf{z}$. With the fixed matrix $\mathbf{X}$ and vector $\mathbf{z}$, we can explicitly define function $\mathcal{G}$ as (see Rasmussen (2004) for detail)

$$\mathcal{G}(\mathbf{x}) = \mathbf{k}_{\mathbf{x}\mathbf{X}}(\mathbf{K}_{\mathbf{X}\mathbf{X}})^{-1}\mathbf{L}\mathbf{z},$$

where $\mathbf{k}_{\mathbf{x}\mathbf{X}}$ is the covariance vector between $\mathbf{x}$ and $\mathbf{X}$, $\mathbf{K}_{\mathbf{X}\mathbf{X}}$ is the covariance matrix between $\mathbf{X}$ and $\mathbf{X}$, and $\mathbf{L}$ is the lower Cholesky decomposition of $\mathbf{K}_{\mathbf{X}\mathbf{X}}$.

We further define the physical model as $\mathcal{F}^p(\mathbf{x}) = \mathcal{G}(\mathbf{x}) + \sqrt{\mathbf{a}^T\mathbf{x}}$, where $\mathbf{a}^T = [0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 1.9]$ is a vector of coefficients. By definition of $\mathcal{F}^s(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\psi})$ and $\mathcal{F}^p(\mathbf{x})$, we subsequently get $\mathcal{F}^{\boldsymbol{\theta}}(\mathbf{x}) = \sqrt{\mathbf{a}^T\mathbf{x}}$.

To form the computational and physical datasets, we locate $m = 30$ control vectors, $\mathbf{x}^p$, uniformly on the square $[0,1]^{10}$. Then for each $\mathbf{x}^p$, we sample 10 functional calibration vectors randomly from the interval $[0,3]$; therefore, we have a total of $n = 300$ computational data points. Finally, we sample 100 random test control vectors, $\mathbf{x}^*$, from the same square $[0,1]^{10}$ to form a test dataset. Table 5 shows the performace of BNMC, NBC, NFC, and PFC in terms of RMSE and computation time.

Methodology	RMSE_y	RMSE_θ	Computation time (in seconds)
NFC	0.89	0.21	60.19
PFC	0.89	0.24	1595.03
NBC	0.94	1.62	663.57
BNMC	0.90	0.18	1256.5

Table 5: RMSE_y, RMSE_θ, and computation time of different methodologies for the 10 dimensional problem

Due to the fact that the input space is large, and we are using a small number of samples, none of the competing methodologies have an advantage over the others in terms of RMSE_y; however, BNMC outperforms the other methodologies in terms of RMSE_θ. However, when we consider both RMSE_y and time, NFC overall performs the best among competing methods.

Appendix E Analysis of residuals

This section conducts statistical analysis on the residuals of the problems discussed in Section 6 to validate the assumptions made for the proposed model in equation (3). We present both visual inspection and statistical testing to validate the assumptions that the errors, ϵ_i^p, are independent and identically distributed with zero mean and constant variance. However, for the assumption of constant variance, due to the limited number of data points in each dataset, we cannot perform any formal statistical test. This is because such statistical tests for equality of variance either require inherent categorical structures or a sufficient number of samples for bucketization.

We present the results for the simulated datasets in Figure 6 and for real datasets in Figure 7. To check the normality assumption, we first plot the normal probability plots of the residuals. As shown in the right panels of Figures 6 and 7 (that is, Figures 6b, 6d, 6f, 7b, and 7d), residuals mostly lie on the diagonal line that represents the theoretical normal distribution. We further perform the Wilk-Shapiro test (Shapiro and Wilk, 1965) to formally test the normality. The first column of Table 6 shows the p-value of the Wilk-Shapiro test (with H_0 : residuals are normally distributed) on the residuals of all the five problems. All the p-values are greater than the significance level of 0.05 suggesting that there is no strong evidence against the assumption of the residuals being normally distributed.

To validate the assumption of zero mean, we first plot the residuals against their fitted values for each problem. As shown in the left panels of Figures 6 and 7 (that is, Figures 6a, 6c, 6e, 7a, and 7c), residuals are spread around the horizontal zero line. We further perform a t-test to formally test the zero mean assumption. The second column of Table 6 shows the p-value of the t-test (with H_0 : residuals have zero mean) on the residuals of all the five problems. All the p-values are greater than the significance level of 0.05 meaning that there is no strong evidence against the zero mean assumption.

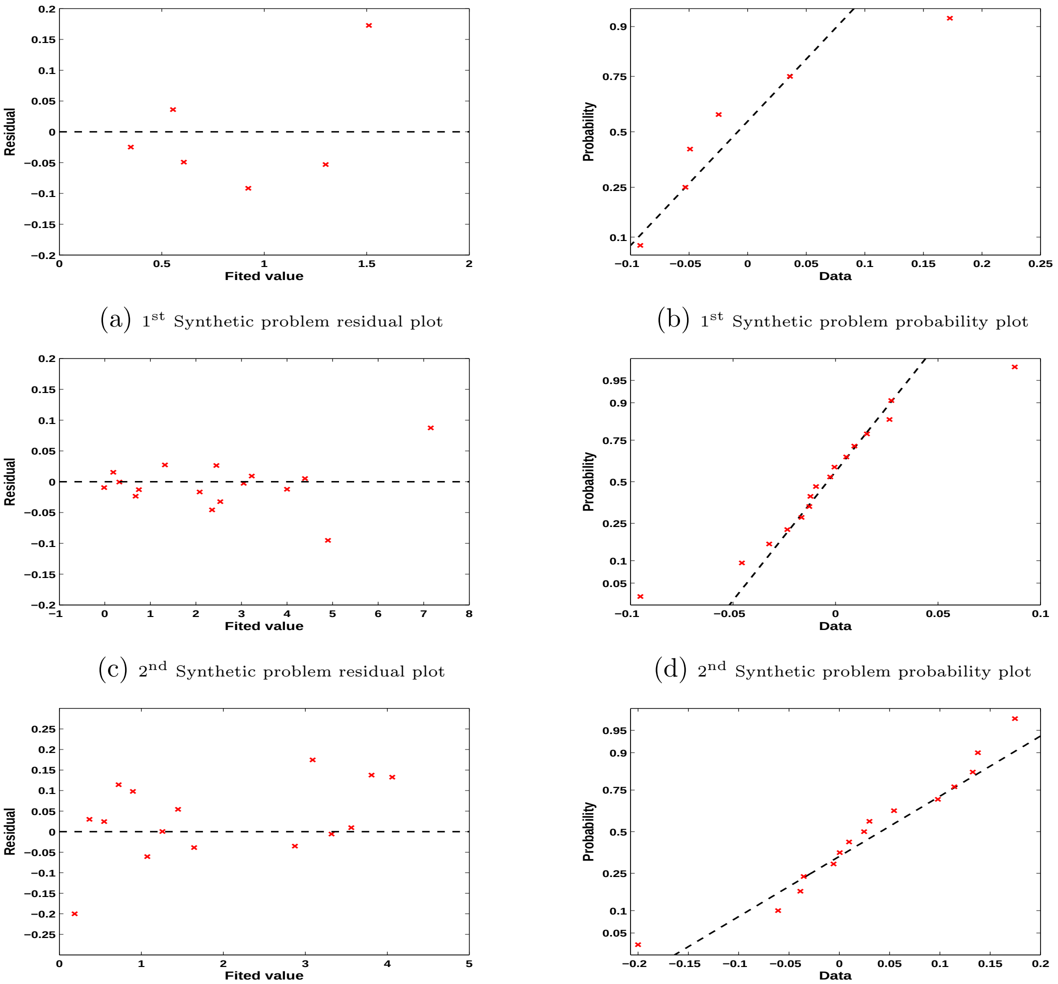
Finally we can visually inspect in Figures 6a, 6c, 6e, 7a, and 7c that there is no clearly observable dependence between the residuals when they are ordered against their fitted values. To formally test this assumption, we order the residuals of each problem against each of their control variables as well as their fitted values. Then, for each order, we conduct the LjungBox test (Ljung and Box, 1978) of lag-1 auto-correlation (with H_0 : lag-1 auto-correlation is zero). As shown in Table 7, all of the p-values of LjungBox test are greater than the significance level of 0.05 meaning that there is no strong evidence against lag-1 auto-correlation being equal to zero.

Problem	Wilk-Shapiro test of normality	t-test for zero mean
1$^{\text{nd}}$ synthetic problem	0.8571	0.967
2$^{\text{nd}}$ synthetic problem	0.9364	0.6113
3$^{\text{nd}}$ synthetic problem	0.9513	0.2591
PVA	0.9282	0.7019
Spot Welding	0.3044	0.3303

Table 6: p-values for Wilk-Shapiro test of normality and t-test for zero mean conducted on the residuals of all the five problems

Problem	LjungBox test for independence ordered against first control variable	LjungBox test for independence ordered against second control variable	LjungBox test for independence ordered against third control variable	LjungBox test for independence ordered against fitted values
1^{nd} synthetic problem	0.5359	0.6931	-	-
2^{nd} synthetic problem	0.6905	0.5529	0.3947	-
3^{nd} synthetic problem	0.2247	0.2247	-	-
PVA	0.3044	0.3044	-	-
Spot Welding	0.3261	0.2446	0.4309	0.084

Table 7: p-values for LjungBox test for lag-1 auto-correlation conducted on the residuals of all the five problems



(a) 1st Synthetic problem residual plot

(b) 1st Synthetic problem probability plot

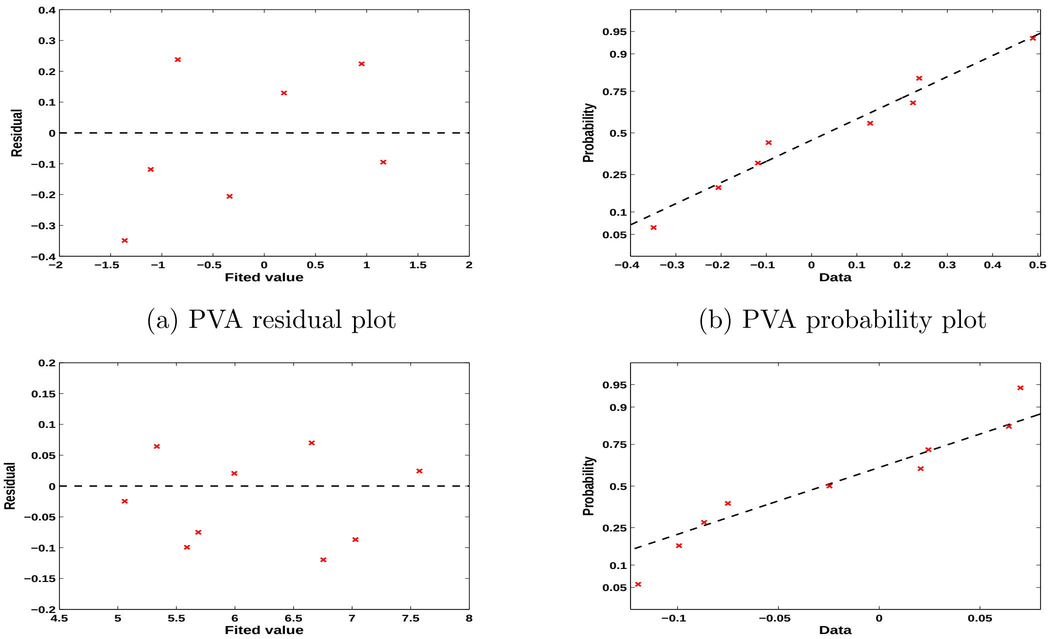
(c) 2nd Synthetic problem residual plot

(d) 2nd Synthetic problem probability plot

(e) 3rd Synthetic problem residual plot

(f) 3rd Third synthetic problem

Figure 6: Figures on the left side of the panel show the residuals against their predicted values and the figures on the right side of the panel show normal probability plots of residuals for the synthetic problems discussed in Section 6



(a) PVA residual plot

(b) PVA probability plot

(c) Spot welding residual plot

(d) Spot welding probability plot

Figure 7: Figures on the left side of the panel show the residuals against their predicted values and the figures on the right side of the panel show probability plot of residuals for real problems discussed in Section 6