Constructing a simulation surrogate with partially observed output

Moses Y-H. Chan, Matthew Plumlee*
Department of Industrial Engineering and Management Sciences,
Northwestern University
and
Stefan M. Wild†
Applied Mathematics and Computational Research Division,
Lawrence Berkeley National Laboratory
NAISE, Northwestern University

April 10, 2023

Abstract

Gaussian process surrogates are a popular alternative to directly using computationally expensive simulation models. When the simulation output consists of many responses, dimension-reduction techniques are often employed to construct these surrogates. However, surrogate methods with dimension reduction generally rely on complete output training data. This article proposes a new Gaussian process surrogate method that permits the use of partially observed output while remaining computationally efficient. The new method involves the imputation of missing values and the adjustment of the covariance matrix used for Gaussian process inference. The resulting surrogate represents the available responses, disregards the missing responses, and provides meaningful uncertainty quantification. The proposed approach is shown to offer sharper inference than alternatives in a simulation study and a case study where an energy density functional model that frequently returns incomplete output is calibrated.

Keywords: Gaussian process, missing data, high-dimensional output, statistical emulation, calibration

*This material is based upon work supported by NSF grants OAC 2004601, DMS 1953111, 1952897.

†This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research SciDAC and applied mathematics programs under contract DE-AC02-05CH11231, and by NSF grants OAC 2004601 and DMS 1952897.

1 Introduction

Computer simulations are used to understand and analyze systems where directly experimenting on the real system is difficult or infeasible. The output of a simulation model depends on a user-specified input configuration that defines the physical and controllable properties of the system. When a user simulates the system, also referred to as running the simulation, they receive outputs related to quantities of interest to the user. Running a simulation can be computationally expensive; each run can cost thousands of core-hours, see for examples the simulation of storm surge (Plumlee et al., 2021a), influenza spread (Venkatramanan et al., 2021), and nuclear dynamics (Phillips et al., 2021). Because these simulations are expensive, it is often helpful to build an emulator, or "surrogate," trained on simulation data to predict at unsimulated (i.e., out-of-sample) configurations.

Surrogate technology is often deployed for calibration (Kennedy and O'Hagan, 2001), where an input configuration is represented by a multidimensional parameter. When run at a parameter, the simulation returns a high-dimensional output consisting of multiple responses collected on a set of fixed locations. While there are many variations of the exact type of inference (Tuo and Wu, 2015; Gramacy et al., 2015; Plumlee, 2017), the overall goal is to learn the parameters by aligning the physical observations with the simulation outputs using computational tools like Markov Chain Monte Carlo (MCMC). Because simulation runs are expensive, it is not possible to run the simulation the millions of times needed for MCMC. Instead, the idea is to run the simulation model for a set of representative parameters and to collect the simulation output from each run; consequently, the surrogate is constructed as the prediction conditional on the simulation output data. Important to solving the calibration problem is that the surrogate produces a measure of uncertainty in the surrogate's predictions.

The most prominent tool for building statistical surrogates involves Gaussian processes (GPs)

(Santner et al., 2018; Gramacy, 2020). GPs offer both an accurate prediction and a measure of uncertainty. The case of high-dimensional outputs leads to the computational intractability of many surrogate construction tools. Two approaches have been proposed to remedy this computational challenge. The first one employs a Kronecker structure of the covariance function of the GP, which assumes a separation between parameters and locations (Rougier, 2008; Hung et al., 2015; Guillas et al., 2018; Marque-Pucheu et al., 2020). The second approach employs a dimension-reduction step for the outputs before the building of surrogates (Bayarri et al., 2007; Higdon et al., 2008; Gu and Xu, 2020). This is in contrast to data-reduction procedures that seek to choose a smaller set of points to represent the entire parameter space (e.g., the selection of support points (Mak and Joseph, 2018)). Examples of dimension-reduction procedures include extractions of principal components (Higdon et al., 2008; Lawrence et al., 2017; Gu and Xu, 2020), wavelet coefficients (Bayarri et al., 2007), and calibration-optimal bases (Salter et al., 2019) from the simulation output data. (Salter et al. (2019) require the knowledge of physical observations, in addition to the simulation output.) These procedures require complete data, meaning for each run, the entire output has to be returned by the simulation. However, in seeking high levels in both performance and fidelity, modern simulation codes may produce partially observed outputs. Hung et al. (2015) have developed an EM algorithm to address the issue of partially observed output prior to constructing a surrogate, and their method is included for comparison in this article (see Section 5). In this article, we focus on extending the second approach to incorporate partially observed outputs.

The presence of partially observed output can be attributed to various causes of code failures. One cause is failure in parallel computing environments where separate computations are scattered over a large number of computing nodes. If a computing node experiences failure during calculations, only some of the calculations may be completed. Another cause is related to numerical calculations embedded in simulation codes. For example, in a simulation that involves solving a

system of nonlinear equations, if the system corresponding to a response is inconsistent or particularly ill-conditioned, then no meaningful solution may be found numerically. Another cause comes from simulations where some responses in an output are undefined. It is not always easy to identify a single underlying cause. Consider a parallel computing environment where a response is not returned because a simulation is terminated by the environment when it exceeds a time limit. The response could be missing because the time limit was set too low, but it could also be missing because the numerical calculations within the simulation would never have terminated. Regardless of the cause, the presence of partially observed outputs renders most surrogate methods inapplicable. Partially observed outputs can be viewed in the context of data missingness. There are several classical categories of missingness mechanisms considered by statisticians: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). In practice the underlying causes of partially observed output are difficult to untangle because of the typically deterministic nature of computer code. Failure in computing nodes seems as though it can be considered MCAR, but in many cases it is MAR because run time depends on the input and longer simulations are more likely to encounter a failure than a shorter one. There are many examples of missingness in simulation codes, including climate studies (Chang et al., 2014; Ma et al., 2022); fluid dynamics (Huang et al., 2020); nuclear physics (Bollapragada et al., 2021); and weather dynamics (Plumlee et al., 2021a).

Despite the common occurrence of partially observed output, there are few methods available for high-dimensional surrogate construction in this setting. The naïve approach would be to simply discard the dimension-reduction step entirely and treat each response separately. In this case, one discards the missing responses and pushes forward with surrogate construction with the available responses. The naïve approach includes locations as additional input dimensions. This inclusion allows for the correlation structures among both parameters and locations to be modeled, similar

to the inclusion of time index as an additional input (Bayarri et al., 2009). However, this approach can easily exceed the limit of standard GP inference when the number of data points goes above several thousands. The only recourse while staying with GP inference then leaves GP approximations, which are still significantly more expensive than dimension-reduction approaches and lead to decreased accuracy. Another approach is to neglect correlations between locations and build an independent surrogate for each location. When the output contains a small number of responses, building separate surrogates often suffices (Baker et al., 2022). However, as the number of responses increases, this approach is prohibited by its computational burden. Simplistic imputation is yet another option, where one imputes the missing values and then builds a surrogate using dimension-reduction tools (e.g., Plumlee et al. (2021a)). The imputation process can be done in various ways besides prediction, such as a "constant-liar" imputation used in optimization (e.g., Chevalier and Ginsbourger (2013)). For purposes of uncertainty quantification, these methods are dangerous as they will interpolate the imputed values with zero residual uncertainty. For example, if the entire output from a run is missing, the imputation approach would simply interpolate the imputed values instead of representing them as missing, which seems contrary to the goals of a surrogate to faithfully represent the predictive uncertainty.

In this article we propose a new GP surrogate method that operates well in the presence of partially observed simulation output. The new method retains the computational efficiency found with dimension-reduction methods like those described in Higdon et al. (2008). However, in contrast to previous such approaches, the new method is not limited to complete data. This method involves two components: the imputation of missing values in the data using the principal components and an adjustment to the covariance matrix used in GP prediction. The adjustment to the covariance matrix ensures that one does not interpolate the imputed values at the missingness locations. The resulting surrogate permits the use of data with partially observed output, and

it has two appealing properties: (i) In the case where complete data is observed for a run, the resulting surrogate will interpolate those results and (ii) in the case where no data is observed for a run, the associated data row (even though imputed) will be ignored. The new surrogate method demonstrates robustness empirically under various missingness mechanisms, and provides improved uncertainty quantification in calibration.

The organization of this article is as follows. Section 2 describes the Fayans energy density functional (EDF), the simulation model that motivates this surrogate development. Section 3 introduces the setting and notation employed, and reviews the principal component GP method for surrogate construction. Section 4 details the imputation and covariance adjustment components in the new surrogate method alongside its properties. Section 5 details a numerical experiment to illustrate the surrogate performance under multiple missingness scenarios. Section 6 discusses the calibration of the Fayans EDF model. Section 7 provides further discussions and closes the article.

2 Fayans energy density functional

The development of our proposed surrogate method is motivated by problems such as the calibration of the Fayans EDF (Fayans, 1998; Fayans et al., 2000). The development and refinement of EDF models have proven effective in understanding atomic nuclei (Reinhard and Nazarewicz, 2017). The EDF model investigated in this article was developed by S. A. Fayans and collaborators for describing ground-state properties of nuclei (Fayans, 1998; Fayans et al., 2000), and has since been used for nuclei predictions (e.g., see Yu and Bulgac (2003); Reinhard and Nazarewicz (2017)). Bollapragada et al. (2021) provides a full description of the numerical implementation under consideration in this article. The model takes a 13-dimensional parameter as input and outputs 198 responses. Each response corresponds to a spherical, ground-state, even-even nuclear configuration and an observable class. A total of 72 nuclear configurations and 9 observable classes are consid-

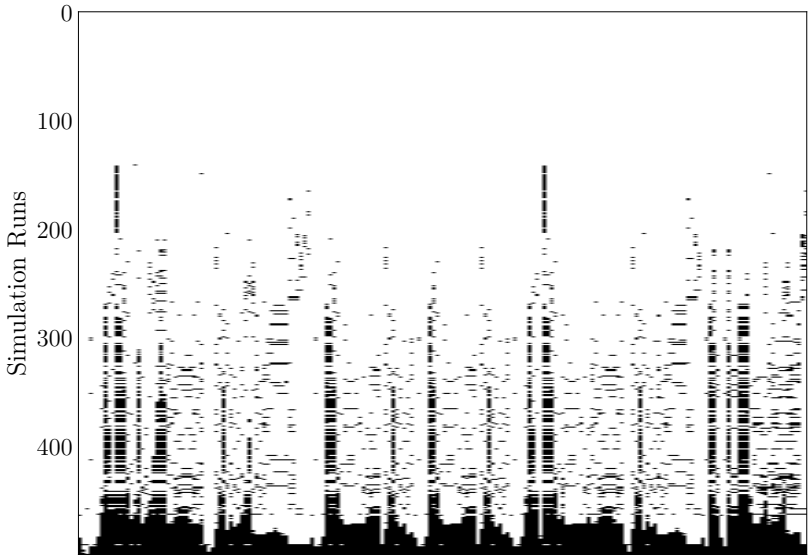

Figure 1: Illustration of partially observed output in our case study (Section 6). Each horizontal line (1–500) is a length 198 simulation output. A dark patch indicates where a response is missing. The horizontal lines are sorted by number of missing values in the output.

ered, totalling to the 198 responses. The list of responses can be found in Table 1 of Bollapragada et al. (2021).

The main relevant feature of this model is that it occasionally produces partial outputs, i.e., within one output, some out of the 198 responses are missing due to code failures, presenting a challenge in constructing a surrogate effectively. Missingness often occurs because an iterative method for solving equations is used in model evaluation, and failure is reported when the method fails to achieve a prescribed accuracy tolerance within an allotted internal iteration budget. Bollapragada et al. (2021) outline the intricacies of possible failures and show that it is not likely the output is MCAR. However, no systematic missingness mechanism is proposed. In previous analyses of another EDF model, $\approx 9\%$ of the model outputs were discarded due to such types of failures (Higdon et al., 2015).

In Section 6 we use a dataset of 500 parameters to construct a surrogate and calibrate the Fayans EDF model. An illustration of this is presented in Figure 1, where out of 500 runs of the model, near 60% (359/500) have at least one missing value.

In calibrating the Fayans EDF model, Bollapragada et al. (2021) have previously employed a point minimization of the total mean-squared loss, or χ^2 loss, with respect to the parameters. Other works calibrated other EDFs in a similar manner, with minimization of the χ^2 loss (Dobaczewski and Olbratowski, 2005; Kortelainen et al., 2010, 2012, 2014; Reinhard and Nazarewicz, 2017). Such approaches do not directly result in the uncertainty quantification sought by nuclear physicists (Dobaczewski et al., 2014). For some other EDFs, uncertainty quantification was performed under a Bayesian framework by Higdon et al. (2015) and McDonnell et al. (2015), but these works assume no missing data (or simply remove output with missing data).

The existing literature does not contain a viable methodology to solve problems like this one. One method is proposed in Ma et al. (2022) to extract functional principal components, which originates from the analysis of partially observed longitudinal data. The locations in Ma et al. (2022) are irregularly spaced between output dimensions, producing partial, often sparse, output at each dimension. In our problem, all responses are expected for each parameter, but missing values may arise for some responses because of various code failures. Lawrence et al. (2017) have proposed to extract principal components using complete output data, and project the partially observed output data onto a subset of basis vectors to obtain the weight coefficients. This treatment of partially observed output data is effective when the locations of missing data are regular. Similarly, in Gu and Xu (2020), the output data are modeled to follow a partition of complete and missing data blocks. The complete data block is then used for principal component analysis following Higdon et al. (2008). In their case, the complete data block contains a large proportion of the output data. However, it is not applicable in our setting since our missing values are irregular. Thus the equivalent complete data block retains only a small proportion of the output data, resulting in inaccurate estimation of the principal components. Another method is proposed by Hung et al. (2015), in which an expectation-maximization procedure is developed to tackle the issue of missing

data, coupled with a separable correlation function that reduces the computational cost. The scale of their intended application is quite small, preventing direct application for our intended application. Hung et al. (2015) have studied a problem with 30 runs and only a few runs with missing data; whereas in the Fayans EDF case, the number of simulation runs is 500, and over half of them result in a partially observed output, recall Figure 1. Hung et al. (2015) is revisited in the numerical experiments in Section 5.

3 High-dimensional surrogates

This section introduces the notations used in this article and reviews standard GP modeling and its principal component based extension to high-dimensional surrogate construction (Higdon et al., 2008).

3.1 Setting and notations

We label the user-specified parameter $\boldsymbol{\theta} \in \mathbb{R}^d$. Because surrogate inference is often deployed in calibration settings, $\boldsymbol{\theta}$ is referred to as a "parameter." We assume that for a simulation run at $\boldsymbol{\theta}$, the simulation output $\boldsymbol{f}(\boldsymbol{\theta}) \in \mathbb{R}^m$ is collected over a fixed set of locations, labeled as $\mathcal{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)$, and thus $\boldsymbol{f}(\boldsymbol{\theta}) = (f(\boldsymbol{\theta}, \boldsymbol{x}_1), \ldots, f(\boldsymbol{\theta}, \boldsymbol{x}_m))^\mathsf{T}$, meaning the simulation evaluated at $\boldsymbol{\theta}$ will produce an output vector $\boldsymbol{f}(\boldsymbol{\theta})$ consisting of m elements. Each element in $\boldsymbol{f}(\boldsymbol{\theta})$ is considered an individual response. For example, in the Fayans EDF, the user-specified $\boldsymbol{\theta}$ represents a 13-dimensional parameter and $\boldsymbol{x} \in \mathcal{X}$ defines a nuclear configuration that corresponds to a response where $m = 198$ (see Section 2).

We presume that we have generated some parameters $(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)$ from a designed computer experiment (such as a Latin Hypercube sample (McKay et al., 1979) or optimized variations thereof (Joseph et al., 2015)), and the collection of simulation responses is arranged as the $n \times m$ matrix

$\boldsymbol{F} = (\boldsymbol{f}(\boldsymbol{\theta}_1)^\mathsf{T}, \ldots, \boldsymbol{f}(\boldsymbol{\theta}_n)^\mathsf{T})^\mathsf{T}$. Each row represents the simulation output for one parameter at all locations $\boldsymbol{x} \in \mathcal{X}$. Each column represents the response collected at all parameters in the computer experiment. Since individual responses may be missing, we denote N as the total number of available responses. If the collection has complete data, $N = nm$. For any matrix $\boldsymbol{A}$, we use $\boldsymbol{A}_{i,\cdot}$ to refer to the ith row, $\boldsymbol{A}_j$ to refer to the jth column of $\boldsymbol{A}$. We use $\boldsymbol{A}_{\mathcal{I}\mathcal{J}}$ to denote a submatrix of $\boldsymbol{A}$ with entries from the set of row indices $\mathcal{I} \subseteq \{1, \ldots, n\}$ and column indices $\mathcal{J} \subseteq \{1, \ldots, m\}$. Commas will be used to separate the row and column indices when the notation is ambiguous.

3.2 Gaussian process surrogates

A surrogate is constructed to enable the prediction of output at an unsimulated parameter. A good surrogate is designed to provide a predictive distribution, which can then be converted to point estimates and uncertainty around those estimates. In this article, a surrogate provides a predictive distribution for $\boldsymbol{f}(\boldsymbol{\theta})$ with a mean $\boldsymbol{\mu}(\boldsymbol{\theta})$ and a covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$. GPs offer a path to do exactly this with a multivariate normal predictive distribution (Santner et al., 2018; Gramacy, 2020). GP is a common choice to build a surrogate because of its flexibility and ability to interpolate.

3.2.1 Univariate GP

This section will review the basics behind GP inference. Consider $g(\boldsymbol{\theta})$ as a univariate function that takes $\boldsymbol{\theta}$ as its input. A GP model with mean function $\gamma(\cdot)$, scale parameter λ, and correlation function $\rho(\cdot, \cdot)$ assumes that for any collection of n scalar outputs (e.g., $\boldsymbol{g} = (g(\boldsymbol{\theta}_1), \ldots, g(\boldsymbol{\theta}_n))^\mathsf{T}$), follows a multivariate normal distribution with mean $\boldsymbol{\gamma} = (\gamma(\boldsymbol{\theta}_1), \ldots, \gamma(\boldsymbol{\theta}_n))^\mathsf{T}$ and covariance matrix $\lambda \boldsymbol{R}$ where $\boldsymbol{R} = (\rho(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j))_{i,j=1}^n$. Typical choices for ρ include a squared exponential or a Matérn correlation function (Handcock and Stein, 1993), but this choice does not impact the rationale

of our method and is left open in this article. Using the GP as a surrogate means that the predictive distribution given $\boldsymbol{g}$ at any test parameter $\boldsymbol{\theta}$ is a normal distribution with mean $\gamma(\boldsymbol{\theta}) + \boldsymbol{r}^{\mathsf{T}}(\boldsymbol{\theta})\boldsymbol{R}^{-1}(\boldsymbol{g}-\boldsymbol{\gamma})$ and variance $\lambda\left(\rho(\boldsymbol{\theta},\boldsymbol{\theta}) - \boldsymbol{r}^{\mathsf{T}}(\boldsymbol{\theta})\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{\theta})\right)$, where $\boldsymbol{r}(\boldsymbol{\theta}) = (\rho(\boldsymbol{\theta},\boldsymbol{\theta}_1),\ldots,\rho(\boldsymbol{\theta},\boldsymbol{\theta}_n))^{\mathsf{T}}$ is the correlation between $\boldsymbol{\theta}$ and $(\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_n)$. The standard form of this inference thus requires $\boldsymbol{R}^{-1}(\boldsymbol{g}-\boldsymbol{\gamma})$, which costs typically $O(n^3)$ operations to compute (though approximations do exist).

The computational cost is a significant obstacle that requires consideration while conducting GP inference. It is critical that the construction of the surrogate and predictions via the surrogate should be fast compared to running the simulation itself (Gramacy, 2020). In the high-dimensional case, the total number of responses is $N = nm$. One could use off-the-shelf GP inference, where each response received from the simulation is individually modeled over both θ and x, i.e., the previously mentioned naïve approach, typically uses $O(N^3)$ computations to obtain the necessary matrix inverses and determinants. When N is moderately large (e.g., 10^4), naïvely using GP inference is difficult because of this computational burden. For our case study, $m = 198$ and hence $N \leq 10^4$ would constrain us to roughly n less than about 50. The accuracy of a GP surrogate is tied to n, and the practical results for our case study found that at $n = 50$ the surrogate is not sufficiently accurate. Researchers have considered this setting before and noted that the problem arises when the model output is multivariate with many responses. A high-dimensional output happens when the number of responses m gets large (often above > 20). There are a few choices for high-dimensional GP surrogates (Bayarri et al., 2007; Higdon et al., 2008; Conti and O'Hagan, 2010), with the majority of these methods seeking to reduce the required computations to $O(n^3)$, such that the computational burden effectively depends only on n.

3.2.2 High-dimensional GP surrogates

Higdon et al. (2008) describe a powerful tool for building surrogates in high-dimensional output

settings. Examples of successful applications include nuclear physics (Higdon et al., 2015) and storm surge (Kyprioti et al., 2021) modeling. The method works by leveraging a low-rank representation for the m-dimensional output by applying a singular value decomposition to the matrix $\boldsymbol{F}$. Specifically, this method seeks $\boldsymbol{\Phi}$, where $\boldsymbol{\Phi}$ is an $m \times \kappa$ matrix defined by a set of κ orthonormal (i.e., $\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{\Phi} = \boldsymbol{I}$) m-dimensional basis vectors. These are chosen such that for some reasonable m-dimensional centering vector $\boldsymbol{c}$, we have $\boldsymbol{F} - \mathbf{1}_n\boldsymbol{c}^\mathsf{T} \approx (\boldsymbol{F} - \mathbf{1}_n\boldsymbol{c}^\mathsf{T})\boldsymbol{\Phi}\boldsymbol{\Phi}^\mathsf{T}$, meaning we can approximately recover our simulation outputs using only the $n \times \kappa$ matrix $\boldsymbol{G} = (\boldsymbol{F} - \mathbf{1}_n\boldsymbol{c}^\mathsf{T})\boldsymbol{\Phi}$. The centering vector is often chosen as the mean of each column in $\boldsymbol{F}$. Then, the original data $\boldsymbol{F}$ are nearly recovered by $\mathbf{1}_n\boldsymbol{c}^\mathsf{T} + \boldsymbol{G}\boldsymbol{\Phi}^\mathsf{T}$. The value of κ is typically chosen to offer sufficient recovery of $\boldsymbol{F}$ from $\boldsymbol{G}$. This representation method is especially effective when there is a strong relationship among the responses of the simulation output because then κ can be made small (i.e., $\kappa \ll m$). Suppose $\boldsymbol{g}(\boldsymbol{\theta}) = \boldsymbol{\Phi}^\mathsf{T}(\boldsymbol{f}(\boldsymbol{\theta}) - \boldsymbol{c})$. Then, the surrogate for the simulation is constructed as $\boldsymbol{c} + \boldsymbol{\Phi}\boldsymbol{g}(\boldsymbol{\theta})$. Thus our goal shifts from building a surrogate on an m-dimensional output $\boldsymbol{f}(\boldsymbol{\theta})$ to building a surrogate on a κ-dimensional output $\boldsymbol{g}(\boldsymbol{\theta})$.

Our goal is now to use information in the matrix $\boldsymbol{G}$ to infer on the projected output $\boldsymbol{g}(\boldsymbol{\theta})$ for any $\boldsymbol{\theta}$. Due to the orthogonal construction of $\boldsymbol{\Phi}$, each component of $\boldsymbol{g}(\cdot) = (g_1(\cdot), \ldots, g_\kappa(\cdot))^\mathsf{T}$ is modeled using an independent GP. Each $g_k(\cdot)$ then follows

$$g_k(\cdot) \sim \mathrm{GP}\left(0, \lambda_k \rho_k(\cdot, \cdot)\right), \tag{1}$$

where, for component k, $\lambda_k > 0$ is a scale parameter, $\rho_k(\cdot, \cdot)$ is a correlation function, and we have set the mean to be zero for ease of exposition. With data column $\boldsymbol{G}_k$, under the GP model, prediction for $g_k(\boldsymbol{\theta})$ follows a normal distribution with mean and variance given by

$$\widehat{\mu}_k(\boldsymbol{\theta}) = \boldsymbol{r}_k^\mathsf{T}(\boldsymbol{\theta})\boldsymbol{R}_k^{-1}\boldsymbol{G}_k \text{ and } \sigma_k^2(\boldsymbol{\theta}) = \lambda_k\left(\rho_k(\boldsymbol{\theta}, \boldsymbol{\theta}) - \boldsymbol{r}_k^\mathsf{T}(\boldsymbol{\theta})\boldsymbol{R}_k^{-1}\boldsymbol{r}_k(\boldsymbol{\theta})\right), \tag{2}$$

where $\boldsymbol{r}_k(\boldsymbol{\theta}) = (\rho_k(\boldsymbol{\theta}, \boldsymbol{\theta}_1), \ldots, \rho_k(\boldsymbol{\theta}, \boldsymbol{\theta}_n))^\mathsf{T}$ and $\boldsymbol{R}_k = (\rho_k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j))_{i,j=1}^n$. From this, the surrogate is then defined by

$$\boldsymbol{\mu}(\boldsymbol{\theta}) = \boldsymbol{c} + \boldsymbol{\Phi}(\widehat{\mu}_1(\boldsymbol{\theta}), \ldots, \widehat{\mu}_\kappa(\boldsymbol{\theta}))^\mathsf{T} \text{ and } \boldsymbol{\Sigma}(\boldsymbol{\theta}) = \boldsymbol{\Phi} \begin{pmatrix} \sigma_1^2(\boldsymbol{\theta}) & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \sigma_\kappa^2(\boldsymbol{\theta}) \end{pmatrix} \boldsymbol{\Phi}^\mathsf{T}. \tag{3}$$

We refer to this method of constructing a surrogate as Principal Component Gaussian Process (PCGP). For PCGP methods, $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ is not of full rank when $\kappa < m$, thus the predictive distribution is often degenerate. However, similar to Higdon et al. (2008), when a surrogate is used to facilitate parameter calibration, $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ is summed with a (typically diagonal) strictly positive definite observation error covariance matrix (see Section 6.1). The resulting sum of the matrices is then full rank and thus $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ not being strictly positive definite does not pose a problem.

The inferences in (3) are used to provide the mean and covariance of the surrogate for $\boldsymbol{f}(\boldsymbol{\theta})$. If there are partially observed simulation outputs, there are reasonable ways to approximate the principal component matrix $\boldsymbol{\Phi}$ and the centering vector $\boldsymbol{c}$ (see, e.g., Roweis (1997); Tipping and Bishop (1999)). The details of approximating $\boldsymbol{\Phi}$, an expectation-maximization algorithm inspired by Roweis (1997), are included in the Supplementary Material (Chan et al., 2021). However, partially observed simulation output in $\boldsymbol{F}$ means that $\boldsymbol{G}$ cannot be computed. Take the ith row of the data for example, if $\boldsymbol{f}(\boldsymbol{\theta}_i)^\mathsf{T}$ has one or more missing responses, then $\boldsymbol{G}_{i,\cdot}$ cannot be computed. When m is large, it can often be the case that at least one response can be missing in a row, which leaves the rest of the data in the same row unusable without remedy. We then risk tossing out a large amount of data because of a few failures. Higdon et al. (2008) also commented on the necessity of complete data for the use of PCGP. This leaves us at a crossroads. If we use the PCGP approach, we cannot handle missing data. If we do not adopt the PCGP approach, the surrogate

computation may be impossible due to the explosive increase in required computations. This article explains how one can expand the PCGP surrogate methodology to handle missing data.

4 Fast surrogates with missing data

We now introduce our method for constructing a surrogate for high-dimensional outputs in the presence of missing data. Specifically, we propose an imputation method for the missing observations during the computation of $\boldsymbol{G}$ alongside a novel adjustment of the covariance matrix. The resulting method is computationally inexpensive and therefore the benefit of fast construction using PCGP is retained.

Section 4.1 describes and illustrates the desired properties of the proposed surrogate. Sections 4.2 and 4.3 detail the imputation and covariance adjustment procedures. Section 4.4 shows how the proposed method delivers the desired surrogate properties. Section 4.5 describes the hyperparameter estimation used in the construction of the surrogate. Section 4.6 provides additional justification for the choice of covariance adjustment.

4.1 Desired surrogate properties

In constructing this surrogate, we consider it necessary to have two desirable properties, the recovery of complete data rows and the ignorance of entirely missing data rows. Meaning, we want to nearly interpolate to chosen precision where the output is complete and disregard where the output is entirely missing.

The motivation for these desired properties can be explained through a thought experiment. Let the collected data $\boldsymbol{F}$ be an $n \times m$ matrix, corresponding to simulation outputs at the parameters $(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)$. Let $\boldsymbol{F}$ only have data available in $n_0 < n$ rows, and for these rows, the outputs are complete. Let the $n_0 \times m$ matrix $\boldsymbol{F}^{(0)}$ denote the available data rows; without loss of generality, we

take $\boldsymbol{F}^{(0)}_{i,\cdot} = \boldsymbol{F}_{i,\cdot}$ for all $i \leq n_0$. We assume that all of the remaining rows are entirely missing; this can occur, for example, when the computer running the simulations went down after completing n_0 rows. Suppose we separately construct two surrogates using a proposed method: one with data $\boldsymbol{F}$ and one with data $\boldsymbol{F}^{(0)}$. Let the surrogate with $\boldsymbol{F}$ be described by $(\boldsymbol{\mu}(\boldsymbol{\theta}), \boldsymbol{\Sigma}(\boldsymbol{\theta}))$, and the surrogate with $\boldsymbol{F}^{(0)}$ be described by $(\boldsymbol{\mu}^{(0)}(\boldsymbol{\theta}), \boldsymbol{\Sigma}^{(0)}(\boldsymbol{\theta}))$. The recovery of complete data rows means that the surrogate should nearly interpolate for the rows with complete data; that is, $\boldsymbol{\mu}(\boldsymbol{\theta}_i) \approx \boldsymbol{f}(\boldsymbol{\theta}_i)$ and $\boldsymbol{\mu}^{(0)}(\boldsymbol{\theta}_i) \approx \boldsymbol{f}(\boldsymbol{\theta}_i)$ for all $i \leq n_0$. The only interpolation error that should exist is due to the dimension reduction ($\kappa < m$) representation we have chosen. The ignorance of entirely missing data rows means that the surrogate should not depend on any of the rows $\boldsymbol{f}(\boldsymbol{\theta}_i)^{\mathsf{T}}$ for any i larger than n_0; that is, $\boldsymbol{\mu}(\boldsymbol{\theta}) = \boldsymbol{\mu}^{(0)}(\boldsymbol{\theta})$ and $\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \boldsymbol{\Sigma}^{(0)}(\boldsymbol{\theta})$ for any $\boldsymbol{\theta}$.

Let us see how obvious approaches, namely the naïve and simplistic imputation approaches described in the introduction, fare with respect to these goals. The naïve method, where one treats each data point individually, would interpolate all observed points and ignore the remainder. Thus it meets our goals, but this method becomes computationally intractable in our settings because of the large N problem. The simplistic imputation approach would impute all rows when dealing with $\boldsymbol{F}$ and then interpolate that imputation. This implies it would nearly interpolate all observed rows, but the predictions from the simplistic imputation approach using $\boldsymbol{F}^{(0)}$ and $\boldsymbol{F}$ are inconsistent. This inconsistency implies that throwing out or keeping rows with fully missing output will give different predictions.

The method proposed in this article is able to nearly interpolate complete rows and ignore missing rows similar to the naïve method, yet it remains computationally tractable. While these criteria do not guarantee interpolation of outputs in the partially observed output case, it is our expectation and experience that by matching these two extremes, the predictions in the partially observed case offer notably better predictions than the simplistic imputation approach. We later

justify this with simulation experiments in Section 5.

4.2 Gaussian process-based imputation

We will assume the GPs modeling $g_k(\cdot)$ are stationary, and thus, without loss of generality, let $\rho_k(\boldsymbol{\theta}, \boldsymbol{\theta}) = 1$ for $k = 1, \ldots, \kappa$. Furthermore, this section will assume the centering constant $\boldsymbol{c} = \mathbf{0}$ for ease of exposition. If we consider the relationship that our output is (nearly) $\boldsymbol{\Phi}\boldsymbol{g}(\boldsymbol{\theta})$, its covariance, following (1), is of the form

$$\boldsymbol{\Phi}\mathrm{Cov}(\boldsymbol{g}(\boldsymbol{\theta}))\boldsymbol{\Phi}^{\mathsf{T}} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^{\mathsf{T}}, \tag{4}$$

where $\boldsymbol{\Lambda}$ is the diagonal matrix $\mathrm{diag}(\lambda_1, \ldots, \lambda_\kappa)$, due to the independence among the κ GPs in (1). These values are presumed decided in advance. For example, a reasonable choice deployed in our examples sets $\lambda_1, \ldots, \lambda_\kappa$ to the square of the corresponding singular values from the decomposition of $\boldsymbol{F}$.

The covariance matrix in (4), similar to the one in (3), is not of full rank. Therefore, it is difficult to use (4) because the low-rank nature implies that one can have effectively "fully observed" $\boldsymbol{f}(\boldsymbol{\theta})$ after observing only κ entries. Instead, we will use a full-rank extension of this matrix. Pick $\varepsilon > 0$ such that $\lambda_k > \varepsilon$ for all $k \leq \kappa$. This choice of ε ensures that this covariance matrix extension for $\boldsymbol{f}(\boldsymbol{\theta})$, defined as

$$\boldsymbol{B} = \boldsymbol{\Phi}(\boldsymbol{\Lambda} - \varepsilon\boldsymbol{I})\boldsymbol{\Phi}^{\mathsf{T}} + \varepsilon\boldsymbol{I}, \tag{5}$$

is of full rank, and thus $\boldsymbol{B}$ and any principal submatrix of $\boldsymbol{B}$ are invertible.

Using the covariance matrix in (5), we impute the missing observations to build $\tilde{\boldsymbol{G}}$, an $n \times \kappa$ matrix with complete entries, to replace $\boldsymbol{G}$.

Let $\mathcal{J}(i) \subseteq \{1, \ldots, m\}$ be the set of column indices where data are not missing in $\boldsymbol{f}(\boldsymbol{\theta}_i)^{\mathsf{T}}$. Then, let $\boldsymbol{F}_{i,\mathcal{J}(i)} = (f(\boldsymbol{\theta}_i, \boldsymbol{x}_j))_{j\in\mathcal{J}(i)}^{\mathsf{T}}$ be the row vector that contains the available data for parameter

$\boldsymbol{\theta}_i$ and $\boldsymbol{B}_{\mathcal{J}(i),\cdot}, \boldsymbol{B}_{\mathcal{J}(i),\mathcal{J}(i)}$ be the submatrices corresponding to the indices. Because this section assumes a centering vector of zeros,

$$\begin{pmatrix} \boldsymbol{f}(\boldsymbol{\theta}_i) \\ \boldsymbol{F}_{i,\mathcal{J}(i)}^{\mathsf{T}} \end{pmatrix} \sim N\left(\begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{pmatrix}, \begin{pmatrix} \boldsymbol{B} & \boldsymbol{B}_{\mathcal{J}(i),\cdot}^{\mathsf{T}} \\ \boldsymbol{B}_{\mathcal{J}(i),\cdot} & \boldsymbol{B}_{\mathcal{J}(i),\mathcal{J}(i)} \end{pmatrix} \right),$$

where this is a degenerate multivariate normal distribution. Subsequently the conditional mean and covariance of $\boldsymbol{f}(\boldsymbol{\theta}_i)$ given $\boldsymbol{F}_{i,\mathcal{J}(i)}^{\mathsf{T}}$, following standard normal theory (e.g., Gelman et al. (2013)), are

$$\mathbb{E}(\boldsymbol{f}(\boldsymbol{\theta}_i)|\boldsymbol{F}_{i,\mathcal{J}(i)}^{\mathsf{T}}) = \left(\boldsymbol{F}_{i,\mathcal{J}(i)}\boldsymbol{B}_{\mathcal{J}(i),\mathcal{J}(i)}^{-1}\boldsymbol{B}_{\mathcal{J}(i),\cdot}\right)^{\mathsf{T}} \quad \text{and}$$

$$\mathrm{Cov}(\boldsymbol{f}(\boldsymbol{\theta}_i)|\boldsymbol{F}_{i,\mathcal{J}(i)}^{\mathsf{T}}) = \boldsymbol{B} - \boldsymbol{B}_{\mathcal{J}(i),\cdot}^{\mathsf{T}}\boldsymbol{B}_{\mathcal{J}(i),\mathcal{J}(i)}^{-1}\boldsymbol{B}_{\mathcal{J}(i),\cdot},$$

respectively. We propose then to infer $\boldsymbol{G}_{i,k}$ using the conditional quantities

$$\tilde{\boldsymbol{G}}_{i,k} = \boldsymbol{F}_{i,\mathcal{J}(i)}\boldsymbol{B}_{\mathcal{J}(i),\mathcal{J}(i)}^{-1}\boldsymbol{B}_{\mathcal{J}(i),\cdot}\boldsymbol{\Phi}_{\cdot,k} \quad \text{and} \tag{6}$$

$$u_{ik} = \boldsymbol{\Phi}_{\cdot,k}^{\mathsf{T}}\left(\boldsymbol{B} - \boldsymbol{B}_{\mathcal{J}(i),\cdot}^{\mathsf{T}}\boldsymbol{B}_{\mathcal{J}(i),\mathcal{J}(i)}^{-1}\boldsymbol{B}_{\mathcal{J}(i),\cdot}\right)\boldsymbol{\Phi}_{\cdot,k}. \tag{7}$$

This proposed inference is the key insight for our approach. While (6) is a reasonable use of our principal components for imputation, (7) is a valuable measure of goodness for the imputation, since the conditional variances reflect the imputation uncertainty. The quantities u_{ik} from (7) are used to adjust our GP's covariance matrix to account for the uncertainty due to imputation. For the suggested inferences to be practically useful, the inversion of the submatrix of $\boldsymbol{B}$ should be efficient. The details of such inversion are provided in the Supplementary Material (Chan et al., 2021).

4.3 Covariance adjustment

The additional uncertainty from the imputation of missing data needs to be accounted for. We propose adjusting the covariance matrix by adding an extra term to $\boldsymbol{R}_k$, the original covariance matrix in (2), in order to model the increase in variance of prediction due to imputation. Defining the scaled predictive variances by $w_{ik} = u_{ik}/\lambda_k$, the proposed adjusted covariance matrix is

$$\tilde{\boldsymbol{R}}_k = \boldsymbol{R}_k + \beta_k \text{diag}(v_{1k}, \ldots, v_{nk}), \qquad \text{where } v_{ik} = \min\left\{\eta, \frac{w_{ik}}{(1-w_{ik})^\alpha}\right\}, \tag{8}$$

where $\eta > 0$ is a large constant introduced to prevent infinite values from corrupting our linear algebra, and $\alpha > 0$ and $\beta_k > 0$ are hyperparameters that affect the magnitude of the additional term. In particular, α controls the penalty for extreme missingness in an output and β_k controls the amplification of additional variances for component k. In the case if we set $\beta_k = 0$, the variance added from imputation is not accounted for, and the imputed values are treated as observed and interpolated.

Using (6) on every row with missing data, $\tilde{\boldsymbol{G}}_k$ is the completed kth column of the low-dimensional data. Combining that with our covariance in (8) gives that prediction for $g_k(\theta)$ with

$$\tilde{\mu}_k(\boldsymbol{\theta}) = \boldsymbol{r}_k^\mathsf{T}(\boldsymbol{\theta})\tilde{\boldsymbol{R}}_k^{-1}\tilde{\boldsymbol{G}}_k \quad \text{and} \tag{9}$$

$$\tilde{\sigma}_k^2(\boldsymbol{\theta}) = \lambda_k\left(\rho_k(\boldsymbol{\theta},\boldsymbol{\theta}) - \boldsymbol{r}_k^\mathsf{T}(\boldsymbol{\theta})\tilde{\boldsymbol{R}}_k^{-1}\boldsymbol{r}_k(\boldsymbol{\theta})\right), \tag{10}$$

where $\boldsymbol{v}_k = (v_{1k}, \ldots, v_{nk})^\mathsf{T}$. The completed version of the surrogate provides a prediction for $\boldsymbol{f}(\boldsymbol{\theta})$

that has a multivariate normal distribution with updated mean and covariance

$$\boldsymbol{\mu}(\boldsymbol{\theta}) = \boldsymbol{\Phi}(\tilde{\mu}_1(\boldsymbol{\theta}), \ldots, \tilde{\mu}_\kappa(\boldsymbol{\theta}))^\mathsf{T}, \text{ and } \boldsymbol{\Sigma}(\boldsymbol{\theta}) = \boldsymbol{\Phi} \begin{pmatrix} \tilde{\sigma}_1^2(\boldsymbol{\theta}) & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \tilde{\sigma}_\kappa^2(\boldsymbol{\theta}) \end{pmatrix} \boldsymbol{\Phi}^\mathsf{T}. \tag{11}$$

Recall that for this section we set $\boldsymbol{c} = 0$, thus these terms agree closely with the PCGP terms in (3), where we have only modified the predictions of means and variances. If there is no missingness in our simulation output, the expressions in (3) and (11) will exactly agree.

4.4 Properties of the proposed surrogate

The following result guarantees that if there is no missing data in the ith row (i.e., $\mathcal{J}(i) = \{1, \ldots, m\}$), then $\tilde{\boldsymbol{G}}_{i,\cdot}$ recovers $\boldsymbol{G}_{i,\cdot}$ without additional uncertainty.

Theorem 1 (Recovery of fully observed row)**.** If $\mathcal{J}(i) = \{1, \ldots, m\}$, then $\tilde{\mu}_k(\boldsymbol{\theta}_i) = \boldsymbol{f}(\boldsymbol{\theta}_i)^\mathsf{T} \boldsymbol{\Phi}_{\cdot,k}$ and $\tilde{\sigma}_k^2(\boldsymbol{\theta}_i) = 0$ for $k = 1, \ldots, \kappa$.

The proof is given in the Supplementary Material (Chan et al., 2021). On the other hand, notice that if there is no data in the ith row (i.e., $\mathcal{J}(i)$ is the empty set), then

$$u_{ik} = \boldsymbol{\Phi}_{\cdot,k}^\mathsf{T} \left(\boldsymbol{\Phi}(\boldsymbol{\Lambda} - \varepsilon \boldsymbol{I}) \boldsymbol{\Phi}^\mathsf{T} + \varepsilon \boldsymbol{I} \right) \boldsymbol{\Phi}_{\cdot,k} = \lambda_k - \varepsilon + \varepsilon = \lambda_k. \tag{12}$$

Our next result establishes that our method naturally ignores these rows of missing data.

Theorem 2 (Ignorance of missing rows)**.** Define the kth surrogate component, where all rows with entirely missing data are excluded from construction, by $\underline{\tilde{\mu}}_k$ and $\underline{\tilde{\sigma}}_k^2$ from (9) and (10), respectively. If $\mathcal{J}(i) = \emptyset$, then for any $\alpha > 0$, $\beta_k > 0$, $\boldsymbol{\theta}$, and $k \in \{1, \ldots, \kappa\}$, we have that $\tilde{\mu}_k(\boldsymbol{\theta}) - \underline{\tilde{\mu}}_k(\boldsymbol{\theta}) \to 0$ and $\tilde{\sigma}_k^2(\boldsymbol{\theta}) - \underline{\tilde{\sigma}}_k^2(\boldsymbol{\theta}) \to 0$ as $\eta \to \infty$.

The proof is provided in the Supplementary Material (Chan et al., 2021). Although Theorem 2 is stated in a limit as our numerically stabilizing parameter η goes to infinity, we note that in our deployments on real problems that setting η to 10 results in reasonable ignoring behavior.

4.5 Estimation of hyperparameters in surrogate construction

To construct the surrogate is to fit the hyperparameters for all $k = 1, \ldots, \kappa$ components. Besides the covariance adjustment coefficient β_k, additional hyperparameters are implicitly included in the notation of the covariance function $\rho_k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \rho_k(\boldsymbol{\theta}, \boldsymbol{\theta}'; \boldsymbol{\psi}_k)$, where $\boldsymbol{\psi}_k$ encapsulates the covariance function hyperparameters. Within this section, we use the notation $\tilde{\boldsymbol{R}}_k(\beta_k, \boldsymbol{\psi}_k)$ to highlight the dependence of the adjusted covariance matrix $\tilde{\boldsymbol{R}}_k$ on both β_k and $\boldsymbol{\psi}_k$. The surrogate construction involves the optimization of the log-likelihood with respect to the hyperparameters, which can be simplified to

$$\left(\widehat{\beta}_k, \widehat{\boldsymbol{\psi}}_k\right) = \arg\max_{\beta_k, \boldsymbol{\psi}_k} \Big\{ -\frac{1}{2} \log |\tilde{\boldsymbol{R}}_k(\beta_k, \boldsymbol{\psi}_k)| - \frac{1}{2\lambda_k} \tilde{\boldsymbol{G}}_k^{\mathsf{T}} \tilde{\boldsymbol{R}}_k(\beta_k, \boldsymbol{\psi}_k)^{-1} \tilde{\boldsymbol{G}}_k \Big\}, \tag{13}$$

where $|\boldsymbol{A}|$ returns the determinant of $\boldsymbol{A}$. The evaluation of the log-likelihood involves the decomposition of the covariance matrix that costs $O(n^3)$ operations, where n is the number of parameters. In our deployment, we use a gradient-based L-BFGS optimization solver to approximate the maximum likelihood estimate.

4.6 Rationale for the extra term

We now provide justification for the choice of the extra term in (8) alongside hyperparameter α. While w_{ik} can be used directly as the added variance terms, it is insufficient to represent the intuition that (i) if the row has complete data, no adjustments should be made, and (ii) if no data are observed in the row, the added term should be infinite, because we have no information about

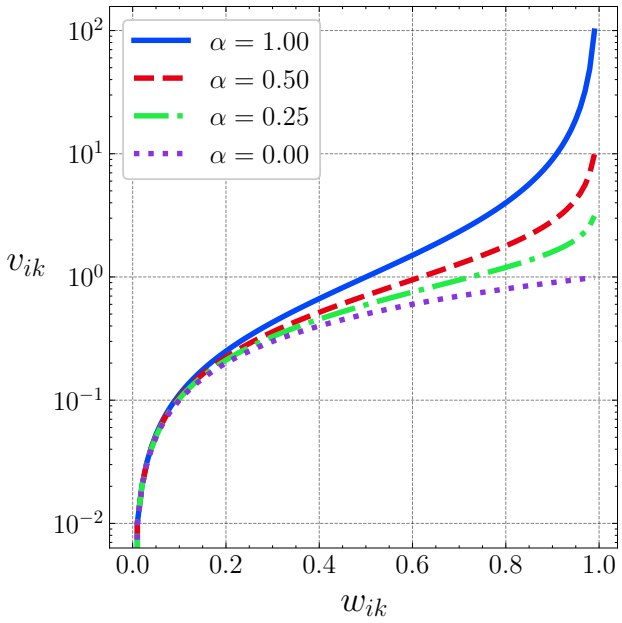

Figure 2: Illustrations of the effect of hyperparameter α in the variance term v_{ik}. Except for $\alpha = 0$, v_{ik} approaches infinity as w_{ik} approaches 1.

the row. In the case where $\alpha = 0$, $v_{ik} = w_{ik}$ does not achieve infinity when no data are observed. When $\alpha > 0$, the added term $v_{ik} = 0$ when $w_{ik} = 0$, and $v_{ik} = \infty$ when $w_{ik} = 1$. Moreover, the magnitude of α controls the rate of increase of variance v_{ik} as w_{ik} increases. The variance term v_{ik} represents the uncertainty introduced from observing only at the index set $\mathcal{J}(i)$ for row i. As α increases, v_{ik} increases with a faster rate to infinity as w_{ik} approaches 1. The effect of α is illustrated in Figure 2. The hyperparameter β_k is introduced to control the inflation (or deflation) of additional variances for the kth surrogate component. The β_k value can vary across the components. When $\beta_k = 0$, the additional variances have no effect to the surrogate, as if there is no uncertainty due to data missingness. When β_k is infinite, the overwhelming additional variances will cause any row with any amount of missing data to be ignored. When $\beta_k = 1$, no adjustment is made. The hyperparameters $\beta_k, k = 1, \ldots, \kappa$, are estimated in the surrogate construction.

A numerical experiment is presented in Supplementary Material (Chan et al., 2021) that investigates the choice of α and if β_k should be either equal to 1 or optimized as a hyperparameter. We

found the best behavior when the β_k are optimized and found $\alpha > 0.5$ could negatively impact the prediction accuracy. We suggest $\alpha = 0.3$ as a default choice.

5 Numerical experiments

We now present the numerical experiment for evaluating the performance of our proposed surrogate method.

5.1 Setup of numerical experiments

We compare the proposed method with other methods that employ common strategies for dealing with missing data. A total of six methods are considered. We label our method as "PCGPwM," for PCGP with Missingness. Among the other methods, the first method ("GP-OM", standing for GP-Omit-Missingness) is to omit the missing data, which is the naïve approach that treats the N responses individually and discards the missing points. The second method ("colGP") is to treat locations as independent and construct a GP for each of m locations. Gu and Berger (2016) refer to colGP as the "Many Single" emulator approach and have commented on its computational intractability when m is large. Gu and Berger (2016) resolves this computation issue by aligning common Gaussian process parameters across locations, but this fix is not needed here as our m is relatively small. We consider the expectation-maximization-based method by Hung et al. (2015) as the third method ("EMGP"). The next two methods involve the simplistic imputation approach with two different imputation procedures. The two imputation procedures considered are the k-nearest neighbor method (see, e.g., Altman (1992)), and Bayesian ridge regression (Tipping, 2001). Since the PCGP method is used on the imputed data, these two methods are termed "PCGP-kNN" and "PCGP-BR," respectively. In addition to these methods, we compare PCGPwM with a baseline method where the principal components are assumed known. This comparison is reported

in the Supplementary Materials (Chan et al., 2021).

GP-OM treats the missing data as if they were not present. We discard any response that is missing. After omitting the missing data, the remaining data do not necessarily form a data matrix. For example, one row may have no missing data, while another row may have one or more entries removed. Instead, the remaining data are stacked into a column and a single GP is constructed as a surrogate. Due to the stacking of data, this method becomes intractable as the computational complexity is $O(N^3)$, with N being the number of available data points. The colGP method treats each $\boldsymbol{x} \in \mathcal{X}$ as independent locations and constructs m GPs. If m is large, computational costs may not permit the use of colGP simply due to the number of GPs to be constructed. The EMGP method estimates the missing values via an EM algorithm and uses a separable covariance matrix to speed up its GP construction. For each of the simplistic imputations, the missing data are first imputed and PCGP is then applied to construct the surrogate. The k-nearest neighbor method imputes the missing value by averaging the k closest available data points, using the Euclidean distance between couples $(\boldsymbol{\theta}, \boldsymbol{x})$. The Bayesian ridge regression fits a regression model and imputes the missing values with the predictions of the model. In the implementation of the imputation methods using `scikit-learn` (Pedregosa et al., 2011), missing values in each data column are imputed in a round-robin fashion for 10 iterations, and the result from the final iteration is returned. See the package documentation on `IterativeImputer` for details.

We use four functions as test examples, namely the borehole function, piston function, wing-weight function, and OTLcircuit function. These are common test functions for emulation and uncertainty quantification, located at `http://www.sfu.ca/~ssurjano/index.html`. The input variables of each function are partitioned into $(\boldsymbol{\theta}, \boldsymbol{x})$. Our proposed method does not address the source of missingness in the data. To test the robustness of the method, we generate partially observed output with the three missing mechanisms mentioned in Section 1. The missing mech-

anisms are MCAR, MNAR, and MAR. Under MCAR, any response has the same probability of missing. Under MNAR, the response missingness depends on unobserved quantities, for example, the value of the response. Under MAR, the response missingness only depends on quantities that are observed, for example, the parameter and location in our case. For each of these missingness mechanisms, the percentage of missingness is specified. The methods are tested at 1%, 5%, and 25% missingness levels. With MNAR, values of the borehole and the wingweight functions are missing when the evaluated function value exceeds a threshold; and the values of the OTLcircuit and the piston functions are missing according to the probability determined by a logistic model over the fixed locations. With MAR, The missingness is randomly assigned over a subset of the fixed locations. The test functions and the details of the MNAR and MAR generation are included in the Supplementary Material (Chan et al., 2021).

The size of the output, m, is set as 15. The set of locations, $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)$, This is determined using $\boldsymbol{x}$'s that are uniformly sampled from their respective ranges. The number of parameters, n, are set to take values $n = 50, 100, 250, 1000$, and 2500. The parameters are sampled using a Latin Hypercube design (Santner et al., 2018) in the unit hypercube, and then scaled to their respective ranges. Denoting a missingness scenario to be the missing mechanism and the missing fraction combined, nine scenarios are considered: (MCAR, 1%), (MCAR, 5%), (MCAR, 25%), (MNAR, 1%), (MNAR, 5%), (MNAR, 25%), (MAR, 1%), (MAR, 5%), and (MAR, 25%). We construct a surrogate using each method, by supplying it the output of the model, for each combination of n and missingness scenario. To efficiently compare the methods, we fix the locations, the parameters, and the set of missing values across the four methods in one replication. Each experiment is run for 20 replications for all test functions. Each replication is given 1 hour of run time and is canceled if the surrogate construction and prediction takes longer than that.

All methods are implemented through the Python package `surmise` (Plumlee et al., 2021b),

a modular package that interfaces different statistical emulation and calibration methods. Our proposed method is implemented under the name `PCGPwM`. For GP-OM, `GPy` (GPy, since 2012) is used to construct the surrogate. The colGP method is implemented under the name `colGP`. The EMGP method is implemented under the name `EMGP`. For PCGP-kNN and PCGP-BR, first the imputations are performed with `scikit-learn` (Pedregosa et al., 2011), then the completed data are supplied to the `PCGP` method to construct a surrogate. The simplistic imputation approach is implemented under the name `PCGPwImpute` with an option of which imputation method to use.

5.2 Results of comparison experiments

This section will discuss representative results for the (MNAR, 5%) missingness scenario. For a full description of the results of our simulation, we direct the reader to the Supplementary Material (Chan et al., 2021).

All methods are competitive in computation, except GP-OM, which costs roughly 30 times as long for all n. Naïvely omitting missing values in the data destroys the regular structure, and further leads to tractability issues. In addition, EMGP does not complete at the largest data size within the time limit. The remaining constructions are completed under the time limit. Note that although the colGP method completes all surrogate constructions within time limit here with $m = 15$, its computation time may be prohibitive as m is too large, further explored in Section 5.3.

The quality of the surrogate methods is measured by the root mean squared error (RMSE), the coverage probability of the 90% prediction interval (90% coverage), and the width of the same interval (90% width). While RMSE concerns the predictive accuracy of the mean, 90% coverage and 90% width are empirical measures of the quality of the surrogate's uncertainty quantification. The measures are evaluated against holdout simulation runs, and any missing values in the runs are excluded from evaluation.

Figure 3 shows the RMSEs for the surrogate methods being compared. For the borehole and the wingweight functions, the RMSE decreases for all methods except the simplistic imputation methods as N increases; whereas for the piston and the OTLcircuit functions, the RMSEs decrease for all methods. GP-OM shows to be generally not competitive in both its computation time and predictive accuracy. The simplistic imputation methods, PCGP-kNN and PCGP-BR, fail to circumvent the issue of missing values, especially when the missingness is MNAR. Since the simplistic imputation approach relies on the availability of close neighbors to the missing points, if the function values are never observed within a certain region, the missing values would be imputed with far-away values of little relevant information. As a result, they result in poor predictions. The EMGP method performs comparably with the simplistic imputation methods, sometimes better, in the case of the wingweight function. The colGP method performs well across all methods in predictive accuracy, especially in the piston and OTLcircuit test functions. The continuous improvement with colGP as N increases shows a potential drawback in using principal-component methods for dimension reduction. Similar conclusions are drawn for (MNAR, 1%) and (MNAR, 25%). We find that in the case of MCAR and MAR, the accuracy of all methods improves as N grows, where PCGPwM performs better than all methods except for colGP.

The 90% coverage records how often the simulation response is contained in the interval produced by the surrogate. The quality of the surrogate is measured by comparing the 90% empirical coverage with the nominal level (in this case, 90%). A coverage close to the nominal level with a narrow width is an indication of a good surrogate. GP-OM results in overcoverage in the borehole function and significantly undercoverage in the remaining functions. By investigating the interval widths, we observe that GP-OM produces a prediction interval too wide in the borehole and too narrow in the others. The EMGP method attains adequate coverage in the piston and wingweight functions, while undercovers in the borehole and wingweight function. As the data size grows,

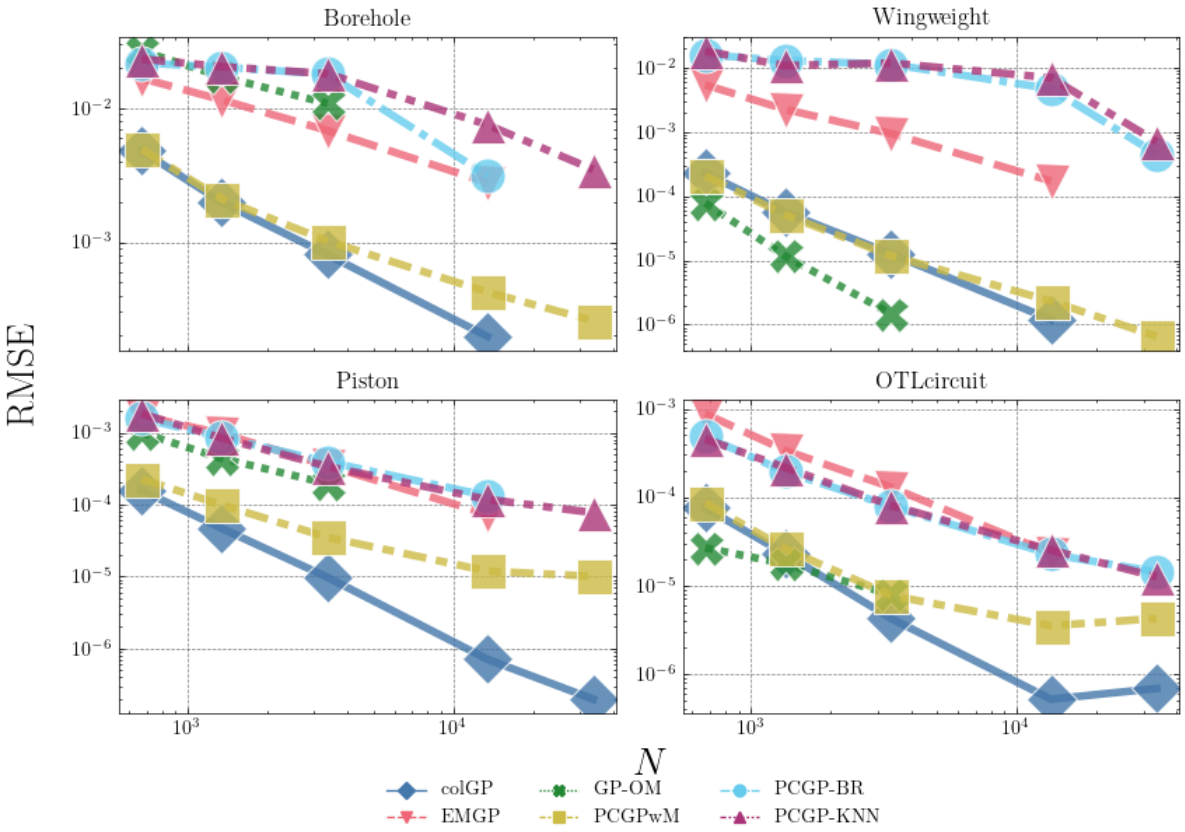

Figure 3: Comparison (log-log scale) of prediction accuracy of surrogate methods for (MNAR, 5%) scenario.

the coverage for the wingweight function increases and attains the prescribed level. The simplistic imputation methods, PCGP-kNN and PCGP-BR, exhibit overcoverages in the borehole and wingweight functions, as a result of wide intervals. PCGPwM achieves the prescribed coverages while being able to provide sharper predictions, indicated by narrower intervals. In the MCAR and MAR scenarios, both the coverage and the prediction interval widths improve as N grows for all methods.

Overall, the proposed PCGPwM method performs generally well in terms of RMSEs and is robust to different types of data missingness. The method also preserves the efficiency of surrogate construction found in dimension-reduction methods such as PCGP (Higdon et al., 2008).

5.3 Additional comparisons of PCGPwM and colGP at higher output dimension

The predictive performance of colGP is laudible but it may be slow to construct at higher output dimensions. To further compare PCGPwM and colGP in terms of computational cost, we have conducted an additional experiment. We have increased the output dimension to $m = 200$ and focused on the performance at larger $n = 1000$, and 2500 with the borehole function. A maximum of 4 hours is permitted for each surrogate construction. Table 1 reports the respective construction time, RMSE, 90% coverage, and 90% width. PCGPwM achieves a higher RMSE than colGP, but maintains the right coverage with a narrower width. At $n = 2500$, the construction time of colGP exceeds the allowed time limit with higher dimensions and the computation is aborted. According to the time scaling, the construction time would have taken colGP an estimated 18 hours.

	Construction time (s)		RMSE ($\times 10^{-4}$)		90% coverage		90% width	
n	PCGPwM	colGP	PCGPwM	colGP	PCGPwM	colGP	PCGPwM	colGP
1,000	640	4,225	3.9	2.5	0.912	0.962	0.764	0.920
2,500	9,680	–	0.62	–	0.981	–	0.571	–

Table 1: Construction times (in seconds) and predictive accuracies of PCGPwM and colGP at output dimension $m = 200$.

6 Case study: Calibrating the Fayans energy density functional

The section describes the case study mentioned in Section 2 that relies on the described surrogate method to conduct calibration. We review a surrogate-based calibration framework in Section 6.1. We present the results of applying the proposed method to the case of calibrating the Fayans EDF in Section 6.2.

6.1 Bayesian calibration with a surrogate

Suppose $\boldsymbol{y} = (y_1, \ldots, y_m)^\mathsf{T}$ is a set of observations from the physical system that the simulation $\boldsymbol{f}(\boldsymbol{\theta})$ is representing. The differences between the observations and the simulation responses are assumed to follow a multivariate normal distribution with zero mean and covariance $\boldsymbol{W}$. This follows the canonical Kennedy and O'Hagan (2001) framework with the assumption that a systematic bias is negligible. The Kennedy and O'Hagan (2001) framework has generated much research interest; for examples, see Higdon et al. (2004); Williams et al. (2006); Bayarri et al. (2007); Higdon et al. (2008); Brynjarsdóttir and O'Hagan (2014).

Let π denote a probability density and $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ be the conditional probability density of $\boldsymbol{\theta}$ given $\boldsymbol{y}$. The purpose for calibration is to infer about the parameter $\boldsymbol{\theta}$. In the Bayesian setting, we are interested in the quantity $\pi(\boldsymbol{\theta}|\boldsymbol{y})$, which is the posterior density of $\boldsymbol{\theta}$. By Bayes rule, given a prior density $\pi(\boldsymbol{\theta})$, we have $\pi(\boldsymbol{\theta}|\boldsymbol{y}) \propto \pi(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$, where $\propto$ denotes equality up to a constant multiplier. Given the distribution of the differences, the expression is expanded to be

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}) \propto |\boldsymbol{W}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{\theta}))^\mathsf{T} \boldsymbol{W}^{-1} (\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{\theta}))\right) \pi(\boldsymbol{\theta}), \tag{14}$$

where $\boldsymbol{W}$ is the covariance matrix of the observation error. Often in a physical experiment, this covariance matrix is diagonal, as in the Fayans EDF model.

Since missing data may be present, $\boldsymbol{f}(\boldsymbol{\theta})$ refers to the hypothetical output responses at a given parameter. When a surrogate, defined by $\boldsymbol{\mu}(\boldsymbol{\theta})$ and $\boldsymbol{\Sigma}(\boldsymbol{\theta})$, constructed with data $\boldsymbol{F}$ is used in place of the simulation model, which is a common strategy in the calibration literature, the posterior density is revised as

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{F}) \propto |\boldsymbol{W} + \boldsymbol{\Sigma}(\boldsymbol{\theta})|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu}(\boldsymbol{\theta}))^\mathsf{T} (\boldsymbol{W} + \boldsymbol{\Sigma}(\boldsymbol{\theta}))^{-1} (\boldsymbol{y} - \boldsymbol{\mu}(\boldsymbol{\theta}))\right) \pi(\boldsymbol{\theta}), \tag{15}$$

by considering the joint distribution of $\boldsymbol{y}$ and $\boldsymbol{F}$ being again multivariate normal. For a fixed simulation model sample $\boldsymbol{F}$ (with missing data), this expression can be used to draw from the posterior using MCMC methods.

6.2 Surrogate construction and calibration for the Fayans EDF

In this section we apply the previously described surrogate methods for the calibration of the Fayans EDF model to provide uncertainty estimates for the parameter, using the knowledge of point minimizers obtained from the previous study. The first method is the proposed PCGPwM method. The second uses the simplistic imputation approach with PCGP-kNN. The third uses the colGP method, and the last method is simply to only use data rows with complete data. GP-OM discussed in the preceding section is not usable in this setting due to the data size causing it to be computationally infeasible: we estimate that the surrogate construction alone would take more than 30 days, before any calibration can be performed. The EMGP method is not applicable either, due to its computational instability. In the original paper by Hung et al. (2015), an isotropic correlation function is used for the categorical variable, which has two levels. In the Fayans EDF model, the categorical variable has nine levels. To adopt the EMGP method, the correlation is chosen to be 1 if the categorical variable is the same, and 0 if not. However, the hyperparameter estimation for the EMGP consistently fails to converge for this application.

We construct the surrogates using simulation outputs of 500 well-spaced parameters. All parameters reside close to a local minimizer of the χ^2 loss from Bollapragada et al. (2021). The parameter space is previously scaled to a unit hypercube such that the dimensions are comparable. To recover the unscaled parameter the centroid of the unscaled hypercube and the scale for each dimension are provided in Table 7 in the Supplementary Material (Chan et al., 2021), which reproduces Table 5 of Bollapragada et al. (2021).

The simulation outputs are partially observed, where approximately 10% of the 99,000 responses are missing. Figure 1, as seen in the introduction, shows the missing value pattern in the sampled outputs arranged by increasing number of failures in parameters. Only 141 rows have complete data.

To calibrate the Fayans EDF, the constructed surrogates are then supplied to the calibrator module in `surmise`. The chosen prior is the Beta$(2, 2)$ distribution over each dimension of the scaled parameter, representing the stability region studied in Bollapragada et al. (2021). The choice Beta$(2, 2)$ reflects the understanding that the parameters closer to the centroid of the scaled hypercube are more plausible. The prior density outside the scaled hypercube is zero, which reflects the boundary of stability region defined by the lower and upper bounds. The prior is then

$$\pi(\boldsymbol{\theta}) \propto \prod_{l=1}^{d} \theta_l \left(1 - \theta_l\right), \boldsymbol{\theta} \in [0, 1]^d,$$

where θ_l is the lth element of $\boldsymbol{\theta}$. With $\pi(\boldsymbol{\theta})$ specified, the posterior is then given by (15). Samples are drawn from the posterior by the Langevin Monte Carlo method (Roberts and Stramer, 2002), an MCMC method that utilizes gradient information at the current iterate. In addition, the sampling method is strengthened by incorporating parallel tempering (Geyer, 1991; Gelman et al., 2013). The method is implemented in the utility module of `surmise` under the name `PTLMC`. We note that the closed-form nature of our surrogate allows for easy deployment of gradient-based approaches.

To investigate the utility of the surrogate-based inference using these methods, we compare the posterior distributions. Since the considered parameter space is scaled around the centroid of the hypercube, we expect the posterior means to be close. This is verified in our computation that the posterior means estimated using all surrogate methods considered are found to be close to each other. We are then concerned with the precision of the posterior, the idea being that a

more precise surrogate in this setting should lead to a narrower posterior on the parameters. This can be seen in expression (15), where a decrease to the surrogate covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ yields a more concentrated posterior, given a fixed $\boldsymbol{\mu}(\boldsymbol{\theta})$. The precision of the surrogate is measured by the widths of the intervals between the 5% and 95% quantiles (called the 90% width) for each parameter relative to the upper and lower bounds. Table 2 contains the 90% widths relatively scaled from all surrogates. The credible intervals from all the surrogates shrink compared with the Beta$(2, 2)$ 90% relative width, 0.730. PCGPwM results in the smallest intervals all but two of the model parameters. This constraining of plausible parameter region is attributed to the improved surrogate offering more precise predictions of simulation responses. The benefit for this case study is that the analysis using the proposed surrogate method provides the uncertainty estimates for the parameter of the Fayans EDF model in contrast to previous studies. We find that there are considerable differences in the resulting interval widths, with some parameters (e.g., f_{ex}^{ξ} and h_{+}^{ξ}) being estimated more precisely, and some parameters (e.g., E/A, K, and L) having smaller precision improvements. To more concretely understand the benefit, if we only use the complete output data to build our surrogate, the resulting posterior intervals would have been 6%–58% wider.

Table 2: Posterior 90% widths relative to their respective ranges for the 13-dimensional parameter using different emulation techniques.

EDF Parameters	PCGPwM	PCGP-kNN	colGP	Complete Data	Prior, Beta$(2, 2)$
ρ_{eq}	**0.355**	0.500	0.546	0.491	0.730
E/A	**0.345**	0.463	0.389	0.607	0.730
K	0.455	0.493	**0.361**	0.643	0.730
J	**0.303**	0.557	0.535	0.497	0.730
L	**0.437**	0.484	0.393	0.576	0.730
h_{2-}^{v}	**0.370**	0.462	0.401	0.587	0.730
a_{+}^{s}	**0.421**	0.569	0.654	0.450	0.730
h_{∇}^{s}	**0.337**	0.405	0.351	0.637	0.730
κ	**0.339**	0.479	0.461	0.614	0.730
κ'	**0.198**	0.319	0.273	0.421	0.730
f_{ex}^{ξ}	**0.125**	0.345	0.246	0.300	0.730
h_{∇}^{ξ}	0.386	0.530	**0.347**	0.536	0.730
h_{+}^{ξ}	**0.128**	0.367	0.215	0.254	0.730

7 Conclusion

This article details a new surrogate construction method developed to handle missing data. The construction relies on an imputation of the missing data and a covariance adjustment to account for the added uncertainty due to imputation. This method is efficient and adds minimal burden on computations. The surrogate construction is effective in ignoring entirely missing data in a multivariate output context. Furthermore, it retains the efficiency of modern approaches to building surrogates of high-dimensional output data.

It is expected that as nascent models are used in large computing environments, partially observed output data will become more prevalent. For example, Lin et al. (in press) conduct their inference by constructing a missingness classifier in addition to a surrogate using available simulation data. When missingness is important to inference, the method proposed in this article could be used in to help with the surrogate to prevent tractability issues.

Acknowledgments

We thank the editor, AE, two anonymous referees, and Earl Lawrence and Kelly Moran for their valuable feedback for improving this article's exposition. We are grateful to Jared O'Neal and Paul-Gerhard Reinhard for developing the Fayans EDF model employed here. We gratefully acknowledge the computing resources provided on Bebop, a high-performance computing cluster operated by the Laboratory Computing Resource Center at Argonne National Laboratory. This research was supported in part through the computational resources and staff contributions provided for the Quest high-performance computing facility at Northwestern University, which is jointly supported by the Office of the Provost, the Office for Research, and Northwestern University Information Technology.

Disclosure statement

The authors report there are no competing interests to declare.

References

Altman, N. S. (1992), "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, 46, 175–185.

Baker, E., Barbillon, P., Fadikar, A., Gramacy, R. B., Herbei, R., Higdon, D., Huang, J., Johnson, L. R., Ma, P., Mondal, A., et al. (2022), "Analyzing stochastic computer models: a review with opportunities," *Statistical Science*, 37, 64–89.

Bayarri, M. J., Berger, J. O., Kennedy, M. C., Kottas, A., Paulo, R., Sacks, J., Cafeo, J. A., Lin, C.-H., and Tu, J. (2009), "Predicting vehicle crashworthiness: Validation of computer models for functional and hierarchical data," *Journal of the American Statistical Association*, 104, 929–943.

Bayarri, M. J., Walsh, D., Berger, J. O., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, J., Parthasarathy, R. J., Paulo, R., and Sacks, J. (2007), "Computer model validation with functional output," *The Annals of Statistics*, 35, 1874 – 1906.

Ben-Ari, E. N. and Steinberg, D. M. (2007), "Modeling data from computer experiments: an empirical comparison of kriging with MARS and projection pursuit regression," *Quality Engineering*, 19, 327–338.

Bollapragada, R., Menickelly, M., Nazarewicz, W., O'Neal, J., Reinhard, P.-G., and Wild, S. M. (2021), "Optimization and supervised machine learning methods for fitting numerical physics models without derivatives," *Journal of Physics G: Nuclear and Particle Physics*, 48, 024001.

Brynjarsdóttir, J. and O'Hagan, A. (2014), "Learning about physical parameters: The importance of model discrepancy," *Inverse Problems*, 30, 114007.

Chan, M. Y. H., Plumlee, M., and Wild, S. M. (2021), "Supplementary material to 'Constructing a simulation surrogate with partially-observed output'," .

Chang, W., Haran, M., Olson, R., and Keller, K. (2014), "Fast dimension-reduced climate model calibration and the effect of data aggregation," *The Annals of Applied Statistics*, 8, 649–673.

Chevalier, C. and Ginsbourger, D. (2013), "Fast computation of the multi-points expected improvement with applications in batch selection," in *Lecture Notes in Computer Science*, Springer, pp. 59–69.

Conti, S. and O'Hagan, A. (2010), "Bayesian emulation of complex multi-output and dynamic computer models," *Journal of Statistical Planning and Inference*, 140, 640–651.

Dobaczewski, J., Nazarewicz, W., and Reinhard, P.-G. (2014), "Error estimates of theoretical models: A guide," *Journal of Physics G: Nuclear and Particle Physics*, 41, 074001.

Dobaczewski, J. and Olbratowski, P. (2005), "Solution of the Skyrme-Hartree-Fock-Bogolyubov equations in the Cartesian deformed harmonic-oscillator basis. (V) HFODD (v2.08k)," *Computer Physics Communications*, 167, 214–216.

Fayans, S. (1998), "Towards a universal nuclear density functional," *Journal of Experimental and Theoretical Physics Letters*, 68, 169–174.

Fayans, S., Tolokonnikov, S., Trykov, E., and Zawischa, D. (2000), "Nuclear isotope shifts within the local energy-density functional approach," *Nuclear Physics A*, 676, 49–119.

Forrester, A., Sobester, A., and Keane, A. (2008), *Engineering Design via Surrogate Modelling: A Practical Guide*, John Wiley & Sons.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013), *Bayesian Data Analysis*, CRC Press.

Gentle, J. E. (2007), *Matrix Algebra*, Springer.

Geyer, C. J. (1991), "Markov chain Monte Carlo maximum likelihood," in *Computing Science and Statistics, Proceedings of the 23rd Symposium on the Interface*, pp. 156–163.

GPy (since 2012), "GPy: A Gaussian process framework in python," `http://github.com/SheffieldML/GPy`.

Gramacy, R. B. (2020), *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*, CRC Press.

Gramacy, R. B., Bingham, D., Holloway, J. P., Grosskopf, M. J., Kuranz, C. C., Rutter, E., Trantham, M., and Drake, R. P. (2015), "Calibrating a large computer experiment simulating radiative shock hydrodynamics," *The Annals of Applied Statistics*, 9, 1141–1168.

Gu, M. and Berger, J. O. (2016), "Parallel partial Gaussian process emulation for computer models with massive output," *The Annals of Applied Statistics*, 10, 1317–1347.

Gu, M. and Xu, Y. (2020), "Fast nonseparable Gaussian stochastic process with application to methylation level interpolation," *Journal of Computational and Graphical Statistics*, 29, 250–260.

Guillas, S., Sarri, A., Day, S., Liu, X., and Dias, F. (2018), "Functional emulation of high resolution tsunami modelling over Cascadia," *Annals of Applied Statistics*, 12, 2023–2053.

Handcock, M. S. and Stein, M. L. (1993), "A Bayesian analysis of kriging," *Technometrics*, 35, 403–410.

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

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

Higdon, D., McDonnell, J. D., Schunck, N., Sarich, J., and Wild, S. M. (2015), "A Bayesian approach for parameter estimation and prediction using a computationally intensive model," *Journal of Physics G: Nuclear and Particle Physics*, 42, 034009.

Huang, J., Gramacy, R. B., Binois, M., and Libraschi, M. (2020), "On-site surrogates for large-scale calibration," *Applied Stochastic Models in Business and Industry*, 36, 283–304.

Hung, Y., Joseph, V. R., and Melkote, S. N. (2015), "Analysis of computer experiments with functional response," *Technometrics*, 57, 35–44.

Joseph, V. R., Gul, E., and Ba, S. (2015), "Maximum projection designs for computer experiments," *Biometrika*, 102, 371–380.

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

Kortelainen, M., Lesinski, T., Moré, J. J., Nazarewicz, W., Sarich, J., Schunck, N., Stoitsov, M. V., and Wild, S. M. (2010), "Nuclear Energy Density Optimization," *Physical Review C*, 82, 024313.

Kortelainen, M., McDonnell, J. D., Nazarewicz, W., Olsen, E., Reinhard, P.-G., Sarich, J., Schunck, N., Wild, S. M., Davesne, D., Erler, J., and Pastore, A. (2014), "Nuclear Energy Density Optimization: Shell Structure," *Physical Review C*, 89, 054314.

Kortelainen, M., McDonnell, J. D., Nazarewicz, W., Reinhard, P.-G., Sarich, J., Schunck, N., Stoitsov, M. V., and Wild, S. M. (2012), "Nuclear energy density optimization: large deformations," *Physical Review C*, 85, 024304.

Kyprioti, A. P., Taflanidis, A. A., Plumlee, M., Asher, T. G., Spiller, E., Luettich, R. A., Blanton, B., Kijewski-Correa, T. L., Kennedy, A., and Schmied, L. (2021), "Improvements in storm surge surrogate modeling for synthetic storm parameterization, node condition classification and implementation to small size databases," *Natural Hazards*, 109, 1349–1386.

Lawrence, E., Heitmann, K., Kwan, J., Upadhye, A., Bingham, D., Habib, S., Higdon, D., Pope, A., Finkel, H., and Frontiere, N. (2017), "The mira-titan universe. II. Matter power spectrum emulation," *The Astrophysical Journal*, 847, 50.

Lin, L., Bingham, D., Broekgaarden, F., and Mandel, I. (in press), "Uncertainty Quantification of a Computer Model for Binary Black Hole Formation," *Annals of Applied Statistics*.

Ma, P., Mondal, A., Konomi, B. A., Hobbs, J., Song, J. J., and Kang, E. L. (2022), "Computer model emulation with high-dimensional functional output in large-scale observing system uncertainty experiments," *Technometrics*, 64, 65–79.

Mak, S. and Joseph, V. R. (2018), "Support points," *The Annals of Statistics*, 46, 2562–2592.

Marque-Pucheu, S., Perrin, G., and Garnier, J. (2020), "An efficient dimension reduction for the Gaussian process emulation of two nested codes with functional outputs," *Computational Statistics*, 35, 1059–1099.

McDonnell, J., Schunck, N., Higdon, D., Sarich, J., Wild, S., and Nazarewicz, W. (2015), "Uncertainty quantification for nuclear density functional theory and information content of new measurements," *Physical Review Letters*, 114, 122501.

McKay, M., Beckman, R., and Conover, W. (1979), "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, 239–245.

Muzellec, B., Josse, J., Boyer, C., and Cuturi, M. (2020), "Missing data imputation using optimal transport," in *Proceedings of the 37th International Conference on Machine Learning*, eds. III, H. D. and Singh, A., vol. 119 of *Proceedings of Machine Learning Research*, pp. 7130–7140.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011), "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, 12, 2825–2830.

Phillips, D. R., Furnstahl, R. J., Heinz, U., Maiti, T., Nazarewicz, W., Nunes, F. M., Plumlee, M., Pratola, M. T., Pratt, S., Viens, F. G., and Wild, S. M. (2021), "Get on the BAND Wagon: a Bayesian framework for quantifying model uncertainties in nuclear dynamics," *Journal of Physics G: Nuclear and Particle Physics*, 48, 072001.

Plumlee, M. (2017), "Bayesian calibration of inexact computer models," *Journal of the American Statistical Association*, 112, 1274–1285.

Plumlee, M., Asher, T. G., Chang, W., and Bilskie, M. V. (2021a), "High-fidelity hurricane surge forecasting using emulation and sequential experiments," *Annals of Applied Statistics*, 15, 460–480.

Plumlee, M., Sürer, O., and Wild, S. M. (2021b), *Surmise Users Manual*.

Reinhard, P.-G. and Nazarewicz, W. (2017), "Toward a global description of nuclear charge radii: Exploring the Fayans energy density functional," *Physical Review C*, 95, 064328.

Roberts, G. O. and Stramer, O. (2002), "Langevin diffusions and Metropolis-Hastings algorithms," *Methodology and Computing in Applied Probability*, 4, 337–357.

Rougier, J. (2008), "Efficient emulators for multivariate deterministic functions," *Journal of Computational and Graphical Statistics*, 17, 827–843.

Roweis, S. (1997), "EM algorithms for PCA and SPCA," in *Proceedings of the 10th International Conference on Neural Information Processing Systems*, MIT Press, NeurIPS'97, pp. 626—632.

Salter, J. M., Williamson, D. B., Scinocca, J., and Kharin, V. (2019), "Uncertainty quantification for computer models with spatial output using calibration-optimal bases," *Journal of the American Statistical Association*, 114, 1800–1814.

Santner, T. J., Williams, B. J., and Notz, W. I. (2018), *The Design and Analysis of Computer Experiments*, Springer New York.

Tipping, M. E. (2001), "Sparse Bayesian Learning and the Relevance Vector Machine," *Journal of Machine Learning Research*, 1, 211–244.

Tipping, M. E. and Bishop, C. M. (1999), "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61, 611–622.

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

Venkatramanan, S., Sadilek, A., Fadikar, A., Barrett, C. L., Biggerstaff, M., Chen, J., Dotiwalla, X., Eastham, P., Gipson, B., Higdon, D., Kucuktunc, O., Lieber, A., Lewis, B. L., Reynolds, Z., Vullikanti, A. K., Wang, L., and Marathe, M. (2021), "Forecasting influenza activity using machine-learned mobility map," *Nature Communications*, 12.

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

Yu, Y. and Bulgac, A. (2003), "Energy density functional approach to superfluid nuclei," *Physical Review Letters*, 90, 222501.

Supplementary material to "Constructing a simulation surrogate with partially observed output"

This document includes the supplementary material to the main article "Constructing a simulation surrogate with partially observed output". The supplementary materials are organized in the following order:

Section A – G: Proofs and technical details for computation.

Section H: Test functions for the numerical experiments.

Section I: Full numerical experiment results.

Section J: Original scaling of the Fayans EDF parameter space.

A Proof of Theorem 1

Proof. First, we note that $\tilde{\boldsymbol{G}}_{i,\cdot} = \boldsymbol{G}_{i,\cdot}$ and $u_{ik} = 0$ for all k because

$$\tilde{\boldsymbol{G}}_{i,\cdot} = \boldsymbol{\Phi}^{\mathsf{T}} \boldsymbol{B}^{\mathsf{T}}_{\mathcal{J}(i),\cdot} \boldsymbol{B}^{-1}_{\mathcal{J}(i),\mathcal{J}(i)} \boldsymbol{F}^{\mathsf{T}}_{i,\mathcal{J}(i)} = \boldsymbol{\Phi}^{\mathsf{T}} \boldsymbol{B} \boldsymbol{B}^{-1} \boldsymbol{F}^{\mathsf{T}}_{i,\cdot} = \boldsymbol{G}_{i,\cdot},$$

and for all k,

$$u_{ik} = \boldsymbol{\Phi}^{\mathsf{T}}_{\cdot,k} \left(\boldsymbol{B} - \boldsymbol{B} \boldsymbol{B}^{-1} \boldsymbol{B}^{\mathsf{T}} \right) \boldsymbol{\Phi}_{\cdot,k} = 0.$$

Then we observe that$\boldsymbol{r}_k(\boldsymbol{\theta}_i)$ is the i row of $\boldsymbol{R}_k$, and thus $\boldsymbol{r}_k^{\mathsf{T}}(\boldsymbol{\theta}_i) \boldsymbol{R}_k^{-1} = \boldsymbol{e}_i$, where $\boldsymbol{e}_i$ is the ith vector of the identity matrix. We conclude that

$$\tilde{\mu}_k(\boldsymbol{\theta}_i) = \boldsymbol{e}_i^{\mathsf{T}} \tilde{\boldsymbol{G}}_{\cdot,k} = \tilde{\boldsymbol{G}}_{ik} = \boldsymbol{G}_{ik} = \boldsymbol{f}(\boldsymbol{\theta}_i)^{\mathsf{T}} \boldsymbol{\Phi}_{\cdot,k},$$

and

$$\tilde{\sigma}_k^2(\boldsymbol{\theta}_i) = \lambda_k \left(\rho_k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_i) - \boldsymbol{e}_i^\mathsf{T} \boldsymbol{r}_k(\boldsymbol{\theta}_i) \right) = \lambda_k \left(\rho_k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_i) - \rho_k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_i) \right) = 0.$$

□

B Boundedness of w_{ik}

Lemma 1. For any $\mathcal{J}(i) \subseteq \{1, \ldots, m\}$, w_{ik} given in expression (7) is bounded between 0 and 1.

Proof. For this proof we will shorten $\mathcal{J}(i)$ to $\mathcal{J}$ and drop the subscript i from w_{ik}. We have that

$$\lambda_k w_k = \boldsymbol{\Phi}_k^\mathsf{T} \left(\boldsymbol{B} - \boldsymbol{B}_{\mathcal{J},\cdot}^\mathsf{T} \boldsymbol{B}_{\mathcal{J}\mathcal{J}}^{-1} \boldsymbol{B}_{\mathcal{J},\cdot} \right) \boldsymbol{\Phi}_k.$$

Without loss of generality, say that $\mathcal{I} = \{1, \ldots, t\}$ and $\mathcal{J} = \{t+1, \ldots, m\}$. The matrix $\boldsymbol{B}$ can then be written as

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{B}_{\mathcal{I}\mathcal{I}} & \boldsymbol{B}_{\mathcal{I}\mathcal{J}} \\ \boldsymbol{B}_{\mathcal{J}\mathcal{I}} & \boldsymbol{B}_{\mathcal{J}\mathcal{J}} \end{pmatrix}.$$

Letting $\boldsymbol{v} = \boldsymbol{\Phi}_{\mathcal{I}k}$ and $\boldsymbol{u} = -\boldsymbol{B}_{\mathcal{J}\mathcal{J}}^{-1} \boldsymbol{B}_{\mathcal{J}\mathcal{I}} \boldsymbol{v}$, we have that

$$\boldsymbol{v}^\mathsf{T} \boldsymbol{B}_{\mathcal{I}\mathcal{J}} \boldsymbol{B}_{\mathcal{J}\mathcal{J}}^{-1} \boldsymbol{B}_{\mathcal{J}\mathcal{I}} \boldsymbol{v} = -\boldsymbol{u}^\mathsf{T} \boldsymbol{B}_{\mathcal{J}\mathcal{J}} \boldsymbol{u} - 2\boldsymbol{u}^\mathsf{T} \boldsymbol{B}_{\mathcal{J}\mathcal{I}} \boldsymbol{v},$$

and conclude that

$$\begin{aligned} \boldsymbol{v}^\mathsf{T} \left(\boldsymbol{B}_{\mathcal{I}\mathcal{I}} - \boldsymbol{B}_{\mathcal{I}\mathcal{J}} \boldsymbol{B}_{\mathcal{J}\mathcal{J}}^{-1} \boldsymbol{B}_{\mathcal{I}\mathcal{J}}^\mathsf{T} \right) \boldsymbol{v} &= \boldsymbol{v}^\mathsf{T} \boldsymbol{B}_{\mathcal{I}\mathcal{I}} \boldsymbol{v} + \boldsymbol{u}^\mathsf{T} \boldsymbol{B}_{\mathcal{J}\mathcal{J}} \boldsymbol{u} + 2\boldsymbol{u}^\mathsf{T} \boldsymbol{B}_{\mathcal{J}\mathcal{I}} \boldsymbol{v} \\ &= \begin{pmatrix} \boldsymbol{v} & \boldsymbol{u} \end{pmatrix} \begin{pmatrix} \boldsymbol{B}_{\mathcal{I}\mathcal{I}} & \boldsymbol{B}_{\mathcal{I}\mathcal{J}} \\ \boldsymbol{B}_{\mathcal{J}\mathcal{I}} & \boldsymbol{B}_{\mathcal{J}\mathcal{J}} \end{pmatrix} \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{u} \end{pmatrix}. \end{aligned} \tag{16}$$

Note that $\boldsymbol{B} = \boldsymbol{\Phi}(\boldsymbol{\Lambda} - \varepsilon \boldsymbol{I})\boldsymbol{\Phi}^\mathsf{T} + \varepsilon \boldsymbol{I}$. By construction, since each diagonal element of $\boldsymbol{\Lambda}$ is larger than ε, $\boldsymbol{B}$ is the sum of two positive-definite matrices and is thus positive definite. Since the right-hand side in (16) is larger than zero, we have that $w_k \geq 0$.

Now, since $\boldsymbol{B}_{\mathcal{J}\mathcal{J}}$ is positive definite, $\boldsymbol{B}_{\mathcal{J}\mathcal{J}}^{-1}$ is positive definite and thus

$$(\boldsymbol{B}_{\mathcal{J},\cdot}\boldsymbol{\Phi}_k)^\mathsf{T} \boldsymbol{B}_{\mathcal{J}\mathcal{J}}^{-1} (\boldsymbol{B}_{\mathcal{J},\cdot}\boldsymbol{\Phi}_k) \geq \boldsymbol{0}.$$

We conclude that

$$\boldsymbol{\Phi}_k^\mathsf{T} \left(\boldsymbol{B} - \boldsymbol{B}_{\mathcal{J},\cdot}^\mathsf{T} \boldsymbol{B}_{\mathcal{J}\mathcal{J}}^{-1} \boldsymbol{B}_{\mathcal{J},\cdot} \right) \boldsymbol{\Phi}_k \leq \boldsymbol{\Phi}_k^\mathsf{T} \boldsymbol{B} \boldsymbol{\Phi}_k.$$

Finally, we note that by the orthogonality of $\boldsymbol{\Phi}$ and letting $\boldsymbol{e}_k$ be the kth vector of the identity matrix, we have that

$$\boldsymbol{\Phi}_k^\mathsf{T} \boldsymbol{B} \boldsymbol{\Phi}_k = \boldsymbol{\Phi}_k^\mathsf{T} (\boldsymbol{\Phi}(\boldsymbol{\Lambda} - \varepsilon \boldsymbol{I})\boldsymbol{\Phi}^\mathsf{T} + \varepsilon \boldsymbol{I})\boldsymbol{\Phi}_k = \boldsymbol{e}_k^\mathsf{T} (\boldsymbol{\Lambda} - \varepsilon \boldsymbol{I}) \boldsymbol{e}_k + \varepsilon = \lambda_k,$$

and thus $w_k \leq 1$. □

C Inverse of matrices with an increasing diagonal

Lemma 2. If the norm of an n-by-n matrix $\boldsymbol{A}$ is bounded (i.e., $\|\boldsymbol{A}\| < L$), then $\lim_{\eta \to \infty} (\boldsymbol{A} + \eta \boldsymbol{I})^{-1} = \boldsymbol{0}$.

Proof. For $\eta \geq L, \|\eta^{-1}\boldsymbol{A}\| < 1$. Consequently, for the partial sum $\boldsymbol{T}_r = \boldsymbol{I} + \sum_{j=1}^{r} (-1)^j \eta^{-j} \boldsymbol{A}^j$, we have that $\lim_{r \to \infty} \boldsymbol{T}_r = (\boldsymbol{I} + \eta^{-1}\boldsymbol{A})^{-1}$ (Gentle, 2007, p.135). Consider the norm of the partial sum,

$$\|\boldsymbol{T}_r\| = \left\| \boldsymbol{I} + \sum_{j=1}^{r} (-1)^j \eta^{-j} \boldsymbol{A}^j \right\| \leq \|\boldsymbol{I}\| + \sum_{j=1}^{r} \eta^{-j} \|\boldsymbol{A}\|^j \leq n - 1 + \frac{1 - \eta^{-(r+1)}}{1 - \eta^{-1}} \to n - 1 + \frac{1}{1 - \eta^{-1}},$$

as $r \to \infty$. The sequence of matrices $\{\boldsymbol{T}_r\}_{r=1}^{\infty}$ uniformly converges over the entries, therefore,

$$\lim_{\eta\to\infty}(\boldsymbol{A}+\eta\boldsymbol{I})^{-1} = \lim_{\eta\to\infty}\lim_{r\to\infty}\eta^{-1}\boldsymbol{T}_r = \lim_{r\to\infty}\lim_{\eta\to\infty}\eta^{-1}\left(\boldsymbol{I}+\sum_{j=1}^{r}(-1)^j\eta^{-j}\boldsymbol{A}^j\right) = \boldsymbol{0}.$$

□

D Proof of Theorem 2

Proof. From Lemma 1 in Supplementary Material B, without loss of generality, partition $\{1,\ldots,n\}$ into $\mathcal{Q} = \{1,\ldots,t\}$ and $\mathcal{S} = \{t+1,\ldots,n\}$, where $w_{ik} < 1$ for $i \in \mathcal{Q}$ and $w_{ik} = 1$ for $i \in \mathcal{S}$. Let $\boldsymbol{Q}_k$ be an n-by-n diagonal matrix with diagonal entries $\left(\frac{w_{1k}}{(1-w_{1k})^{\alpha}},\ldots,\frac{w_{tk}}{(1-w_{tk})^{\alpha}},0,\ldots,0\right)$. Let $\boldsymbol{S}$ be $\begin{bmatrix} 0_{t\times(n-t)} \\ \boldsymbol{I}_{n-t} \end{bmatrix}$. Then, once η is big enough such that for all sets $J \subseteq \{1,\ldots,m\}$ with at least one entry, $\frac{w_{ik}}{(1-w_{ik})^{\alpha}} < \eta$, the adjusted covariance matrix can be rewritten as

$$\boldsymbol{R}_k + \beta_k \mathrm{diag}(\boldsymbol{v}_k) = \boldsymbol{R}_k + \beta_k(\boldsymbol{Q}_k + \eta\boldsymbol{S}\boldsymbol{S}^{\mathsf{T}}).$$

Without loss of generality, let $\beta_k = 1$. We drop the subscript k for the remainder of the proof for brevity. Partition the covariance matrix as

$$\boldsymbol{R}+\boldsymbol{Q}+\eta\boldsymbol{S}\boldsymbol{S}^{\mathsf{T}} = \begin{bmatrix} (\boldsymbol{R}+\boldsymbol{Q})_{\mathcal{Q}\mathcal{Q}} & \boldsymbol{R}_{\mathcal{Q}\mathcal{S}} \\ \boldsymbol{R}_{\mathcal{S}\mathcal{Q}} & \boldsymbol{R}_{\mathcal{S}\mathcal{S}}+\eta\boldsymbol{I} \end{bmatrix}.$$

Recall that for an invertible matrix $\boldsymbol{M} = \begin{bmatrix} \boldsymbol{W} & \boldsymbol{X} \\ \boldsymbol{Y} & \boldsymbol{Z} \end{bmatrix}$, the inverse (Gentle, 2007, p.95) is

$$\boldsymbol{M}^{-1} = \begin{bmatrix} \boldsymbol{W} & \boldsymbol{X} \\ \boldsymbol{Y} & \boldsymbol{Z} \end{bmatrix}^{-1} = \begin{bmatrix} (\boldsymbol{W} - \boldsymbol{X}\boldsymbol{Z}^{-1}\boldsymbol{Y})^{-1} & -\boldsymbol{W}^{-1}\boldsymbol{X}(\boldsymbol{Z} - \boldsymbol{Y}\boldsymbol{W}^{-1}\boldsymbol{X})^{-1} \\ -\boldsymbol{Z}^{-1}\boldsymbol{Y}(\boldsymbol{W} - \boldsymbol{X}\boldsymbol{Z}^{-1}\boldsymbol{Y})^{-1} & (\boldsymbol{Z} - \boldsymbol{Y}\boldsymbol{W}^{-1}\boldsymbol{X})^{-1} \end{bmatrix}.$$

The norm $\|\boldsymbol{R}_{\mathcal{S}\mathcal{S}}\|$ is bounded by the largest singular value of $\boldsymbol{R}_{\mathcal{S}\mathcal{S}}$. Using Lemma 2 in Supplementary Material C,

$$\lim_{\eta\to\infty} (\boldsymbol{R}_{\mathcal{S}\mathcal{S}} + \eta\boldsymbol{I})^{-1} = \boldsymbol{0},$$

$$\lim_{\eta\to\infty} \left(\boldsymbol{R}_{\mathcal{S}\mathcal{S}} + \eta\boldsymbol{I} - \boldsymbol{R}_{\mathcal{S}\mathcal{Q}}(\boldsymbol{R}+\boldsymbol{Q})^{-1}_{\mathcal{Q}\mathcal{Q}}\boldsymbol{R}_{\mathcal{Q}\mathcal{S}}\right)^{-1} = \boldsymbol{0}.$$

By the Woodbury identity (Gentle, 2007, p.221),

$$\begin{aligned}
&\left((\boldsymbol{R}+\boldsymbol{Q})_{\mathcal{Q}\mathcal{Q}} - \boldsymbol{R}_{\mathcal{Q}\mathcal{S}}(\boldsymbol{R}_{\mathcal{S}\mathcal{S}} + \eta\boldsymbol{I})^{-1}\boldsymbol{R}_{\mathcal{S}\mathcal{Q}}\right)^{-1} \\
&= (\boldsymbol{R}+\boldsymbol{Q})^{-1}_{\mathcal{Q}\mathcal{Q}} + (\boldsymbol{R}+\boldsymbol{Q})^{-1}_{\mathcal{Q}\mathcal{Q}}\boldsymbol{R}_{\mathcal{Q}\mathcal{S}}\left(\boldsymbol{R}_{\mathcal{S}\mathcal{S}} + \eta\boldsymbol{I} - \boldsymbol{R}_{\mathcal{S}\mathcal{Q}}(\boldsymbol{R}+\boldsymbol{Q})^{-1}_{\mathcal{Q}\mathcal{Q}}\boldsymbol{R}_{\mathcal{Q}\mathcal{S}}\right)^{-1}\boldsymbol{R}_{\mathcal{S}\mathcal{Q}}(\boldsymbol{R}+\boldsymbol{Q})^{-1}_{\mathcal{Q}\mathcal{Q}} \\
&\to (\boldsymbol{R}+\boldsymbol{Q})^{-1}_{\mathcal{Q}\mathcal{Q}}.
\end{aligned}$$

To summarize, as $\eta \to \infty$,

$$\left(\boldsymbol{R}+\boldsymbol{Q}+\eta\boldsymbol{S}\boldsymbol{S}^{\mathsf{T}}\right)^{-1} \to \begin{bmatrix} (\boldsymbol{R}+\boldsymbol{Q})^{-1}_{\mathcal{Q}\mathcal{Q}} & \boldsymbol{0}_{t\times(n-t)} \\ \boldsymbol{0}_{(n-t)\times t} & \boldsymbol{0}_{(n-t)\times(n-t)} \end{bmatrix}.$$

Therefore, for any length-n vector $\boldsymbol{a}$,

$$\left(\left(\boldsymbol{R}+\boldsymbol{Q}+\eta\boldsymbol{S}\boldsymbol{S}^{\mathsf{T}}\right)^{-1}\boldsymbol{a}\right)_{\mathcal{Q}} \to (\boldsymbol{R}+\boldsymbol{Q})^{-1}_{\mathcal{Q}\mathcal{Q}}\,\boldsymbol{a}_{\mathcal{Q}},$$

$$\left(\left(\boldsymbol{R}+\boldsymbol{Q}+\eta\boldsymbol{S}\boldsymbol{S}^{\mathsf{T}}\right)^{-1}\boldsymbol{a}\right)_{\mathcal{S}}\to\mathbf{0}.$$

Pulling these results through our equations for predictions (i.e., (9) and (10)), we conclude that the limits of $\tilde{\mu}_k(\boldsymbol{\theta})$ and $\tilde{\sigma}_k^2(\boldsymbol{\theta})$ will not depend on any row i with $w_{ik}=1$, and thus the result holds. □

E Estimating $\boldsymbol{\Phi}$

The principal component matrix $\boldsymbol{\Phi}$ is estimated with missing values in the observed simulation output $\boldsymbol{F}$. For a certain row i, the index set $\mathcal{J}(i)$ contains the indices where data are not missing. Denote the missing index set as $\sim\mathcal{J}(i)=\{1,\ldots,m\}\setminus\mathcal{J}(i)$. The principal component matrix $\boldsymbol{\Phi}$ is approximated via Algorithm 1 which mirrors the typical EM structure. The simulation output $\boldsymbol{F}$ is assumed to be scaled to zero mean and unit variance in its columns. The E step is standard, employing conditional equations regarding multivariate normal distribution. However, we note that the M step does not result in the exact optimizer since $\epsilon_{\mathrm{M}}>0$. In our implementation, $\epsilon_{\mathrm{M}}=10^{-5}$ provides reasonable performance and ensures the procedure is numerically stable.

Algorithm 1: EM algorithm for estimating $\boldsymbol{\Phi}$

```
   /* Initialization with column means of non-missing values                    */
 1 for j ∈ {1,...,m} do
 2 |   Let I(j) = {i : j ∈ J(i)}.
 3 |   forall i ∉ I(j) do  f̃_j(θ_i) = (1/|I(j)|) Σ_{i∈I(j)} f_j(θ_i).   /* missing */
 4 |   forall i ∈ I(j) do  f̃_j(θ_i) = f_j(θ_i).                  /* non-missing */
 5 end
   /* EM algorithm                                                              */
 6 while convergence criterion not met do
 7 |   Obtain Φ, Λ via SVD of F̃ = (f̃(θ_1)^T, ..., f̃(θ_n)^T)^T.        /* M step */
 8 |   for i ∈ {1,...,n} do                                              /* E step */
 9 |   |   f̃_{∼J(i)}(θ_i) = Φ_{∼J(i),·}(Λ − εI)Φ^T_{J(i),·} (Φ_{J(i),·}(Λ − εI)Φ^T_{J(i),·} + ε_M I)^{-1} f_{J(i)}(θ_i).
10 |   end
11 end
```

F Computing Inverse of $\boldsymbol{B}_{\mathcal{J}(i),\mathcal{J}(i)}$

The main inferences introduced in (6) and (7) depend on inverting the matrix $\boldsymbol{B}_{\mathcal{J}(i),\mathcal{J}(i)}$. Recall the definition of the covariance matrix

$$\boldsymbol{B} = \boldsymbol{\Phi}\,\mathrm{diag}\left(\lambda_1 - \varepsilon, \ldots, \lambda_\kappa - \varepsilon\right)\,\boldsymbol{\Phi}^{\mathsf{T}} + \varepsilon \boldsymbol{I}.$$

Let $\boldsymbol{\Lambda}_\varepsilon = \mathrm{diag}\left(\lambda_1 - \varepsilon, \ldots, \lambda_\kappa - \varepsilon\right)$. Now consider the inverse of $\boldsymbol{B}$ by using the Woodbury matrix identity,

$$\begin{aligned}
\boldsymbol{B}^{-1} &= \left(\varepsilon \boldsymbol{I} + \boldsymbol{\Phi}\boldsymbol{\Lambda}_\varepsilon \boldsymbol{\Phi}^{\mathsf{T}}\right)^{-1} \\
&= \varepsilon^{-1}\boldsymbol{I} - \varepsilon^{-2}\boldsymbol{\Phi}\left(\boldsymbol{\Lambda}_\varepsilon^{-1} + \varepsilon^{-1}\boldsymbol{I}\right)^{-1}\boldsymbol{\Phi}^{\mathsf{T}} \\
&= \varepsilon^{-1}\boldsymbol{I} - \varepsilon^{-1}\boldsymbol{\Phi}\,\mathrm{diag}\left(\frac{\lambda_1 - \varepsilon}{\lambda_1}, \ldots, \frac{\lambda_\kappa - \varepsilon}{\lambda_\kappa}\right)\boldsymbol{\Phi}^{\mathsf{T}} \\
&= \varepsilon^{-1}\left(\boldsymbol{I} - \boldsymbol{\Phi}\,\mathrm{diag}\left(\frac{\lambda_1 - \varepsilon}{\lambda_1}, \ldots, \frac{\lambda_\kappa - \varepsilon}{\lambda_\kappa}\right)\boldsymbol{\Phi}^{\mathsf{T}}\right).
\end{aligned}$$

The expression above is simple to compute and requires no matrix inversion. But we are concerned about the inverse of the submatrix $\boldsymbol{B}_{\mathcal{J}(i),\mathcal{J}(i)}$, where

$$\boldsymbol{B}_{\mathcal{J}(i),\mathcal{J}(i)} = \varepsilon \boldsymbol{I} + \boldsymbol{\Phi}_{\mathcal{J}(i),\cdot}\,\boldsymbol{\Lambda}_\varepsilon\,\boldsymbol{\Phi}^{\mathsf{T}}_{\mathcal{J}(i),\cdot}.$$

Also by Woodbury identity, we then have

$$\begin{aligned}
\boldsymbol{B}^{-1}_{\mathcal{J}(i),\mathcal{J}(i)} &= \left(\varepsilon \boldsymbol{I} + \boldsymbol{\Phi}_{\mathcal{J}(i),\cdot}\,\boldsymbol{\Lambda}_\varepsilon\,\boldsymbol{\Phi}^{\mathsf{T}}_{\mathcal{J}(i),\cdot}\right)^{-1} \\
&= \varepsilon^{-1}\boldsymbol{I} - \varepsilon^{-2}\boldsymbol{\Phi}\left(\boldsymbol{\Lambda}_\varepsilon^{-1} + \varepsilon^{-1}\boldsymbol{\Phi}^{\mathsf{T}}_{\mathcal{J}(i),\cdot}\,\boldsymbol{\Phi}_{\mathcal{J}(i),\cdot}\right)^{-1}\boldsymbol{\Phi}^{\mathsf{T}}.
\end{aligned}$$

The resulting expression cannot be further simplified since $\boldsymbol{\Phi}^{\mathsf{T}}_{\mathcal{J}(i),\cdot}\boldsymbol{\Phi}_{\mathcal{J}(i),\cdot}$ no longer equals the identity matrix or some simple known form. However, the algebraic manipulation may still be beneficial when we consider only the inverse of the inner κ-by-κ matrix, as opposed to the generally larger matrix $\boldsymbol{B}_{\mathcal{J}(i),\mathcal{J}(i)}$.

G Investigation on α, β_k values

We construct surrogates for the four test functions with MCAR responses at 5%. The details of the test functions are given in Supplementary Material H. The root mean squared error (RMSE) between the function values and the surrogate's predicted values was evaluated with a set of holdout simulation runs. Figure 4 shows the RMSE for selected α values with $\beta_k = 1$ or β_k optimized. The RMSE generally increases as α increases. When β_k's are fixed to be 1, the error increases for larger ranges of α, specifically for the wingweight function, the error continuously increases for $\alpha > 0$. The benefit in including β_k's is shown in the generally lower RMSE achieved. We suggest to include β_k's in the hyperparameter estimation in constructing the surrogate. When β_k's are optimized, smaller α values are preferred.

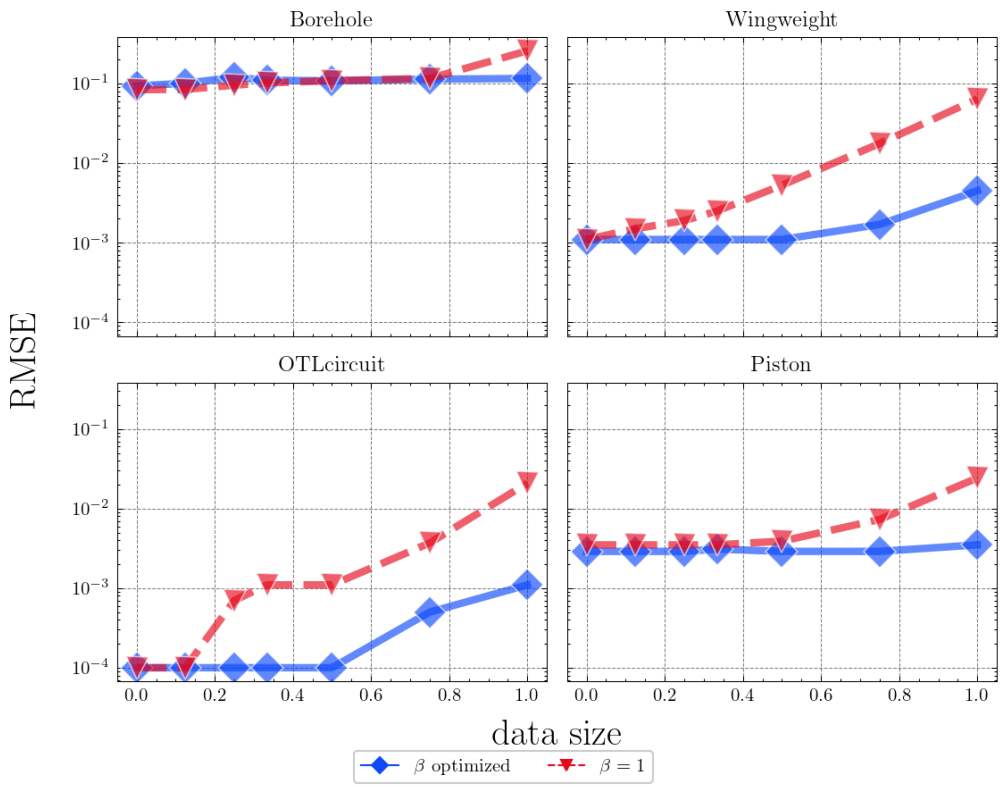

Figure 4: RMSE with $\alpha \in [0, 1]$ and β_k's either optimized or $= 1$.

H Test functions

This section provides details of the four test functions used in evaluating the performance of the algorithm. The subscripts in the expressions in this section refer to the elements in each of the dimension of vector variables $\boldsymbol{\theta}$ and $\boldsymbol{x}$, whereas in all other sections, the subscripted $\boldsymbol{\theta}_i$ and $\boldsymbol{x}_j$ refer to the ith parameter and the jth locations, respectively. The reference parameter $\boldsymbol{\theta}^\star$ refers to the center of the parameter space, for example $(0.5, 0.5, 0.5)^\mathsf{T}$ in a 3D unit hypercube.

For all functions, under MCAR, missingness is assigned with equal probability everywhere with the desired percentage. Under MAR, missingness is randomly assigned to a subset of locations, according to the missing-at-random scheme reported in Muzellec et al. (2020). For MNAR, the missingness mechanism is included in each function below.

H.1 Modified Borehole function

This test function is developed for the purpose of calibration and is modified from the Borehole function (Santner et al., 2018). The function is defined as

$$f(\boldsymbol{\theta}, \boldsymbol{x}) = \frac{2\pi(\theta_1 - x_1)}{2(\theta_2/x_2^2) + \theta_3} \exp\left(\theta_4 x_2\right), \tag{17}$$

where the ranges of the variables are included in Table 3. The first four variables are to be tuned as the parameter, and the last two variables are fixed as locations.

$\boldsymbol{\theta}$	θ_1	[990, 1110]
	θ_2	[0.074, 1.12]
	θ_3	[0.05, 0.5]
	θ_4	[-0.5, 0.5]
$\boldsymbol{x}$	x_1	[700, 820]
	x_2	[0.05, 0.5]

Table 3: Variable ranges for borehole function in (17).

Furthermore, a missingness of MNAR is introduced into the model evaluations, in the following manner:

$$f(\boldsymbol{\theta}, \boldsymbol{x}) \leftarrow \begin{cases} f(\boldsymbol{\theta}, \boldsymbol{x}), & \text{if } f(\boldsymbol{\theta}, \boldsymbol{x}) \leq c_1 f(\boldsymbol{\theta}^\star, \boldsymbol{x}) \\ \text{NA}, & \text{otherwise,} \end{cases} \tag{18}$$

where $\boldsymbol{\theta}^\star$ is the center of the standardized parameter space $U[0,1]^4$. The constant $c_1 > 0$ is adjusted to achieve a desired probability of missingness.

H.2 Piston function

The piston function (Ben-Ari and Steinberg, 2007) is given as

$$f(\boldsymbol{\theta}, \boldsymbol{x}) = 2\pi \sqrt{\frac{M}{k + S^2 \frac{P_0 V_0}{T_0} \frac{T_a}{V^2}}}, \tag{19}$$

where

$$V = \frac{S}{2k}\left(\sqrt{A^2 + 4k\frac{P_0 V_0}{T_0}T_a} - A\right),$$
$$A = P_0 S + 19.62M - \frac{kV_0}{S},$$

and where the variable partition $(\boldsymbol{\theta}, \boldsymbol{x})$ and their ranges are shown in Table 4.

	Variable	Range
$\boldsymbol{\theta}$	k	[1000, 5000]
	P_0	[90000, 110000]
	T_a	[290, 296]
$\boldsymbol{x}$	M	[30, 60]
	S	[0.005, 0.02]
	V_0	[0.0002, 0.01]
	T_0	[340, 360]

Table 4: Variable ranges for the piston function in (19).

The function produces a MNAR that follows the "logistic missing not-at-random" scheme used in Muzellec et al. (2020), where the probability of an entry missing is according to a logistic model over the columns.

H.3 Wingweight function

The wingweight function (Forrester et al., 2008, ch. 1) is given as

$$f(\boldsymbol{\theta}, \boldsymbol{x}) = 0.036 S_w^{0.758} W_{fw}^{0.0035} \left(\frac{A}{\cos^2(\Lambda)}\right)^{0.6} q^{0.006} \lambda^{0.04} \left(\frac{100 t_c}{\cos(\Lambda)}\right)^{-0.3} (N_z W_{dg})^{0.49} + S_w W_p, \quad (20)$$

and the variable partition $(\boldsymbol{\theta}, \boldsymbol{x})$ and the respective ranges are shown in Table 5.

	Variable	Range
$\boldsymbol{\theta}$	A	[6, 10]
	Λ	$[-10°, 10°]$
	q	[16, 45]
	λ	[0.5, 1]
$\boldsymbol{x}$	S_w	[150, 200]
	W_{fw}	[220, 300]
	t_c	[0.08, 0.18]
	N_z	[2.5, 6]
	W_{dg}	[1700, 2500]
	W_p	[0.025, 0.08]

Table 5: Variable ranges for wingweight function in (20).

The function produces a MNAR in the following manner:

$$f(\boldsymbol{\theta}, \boldsymbol{x}) = \begin{cases} f(\boldsymbol{\theta}, \boldsymbol{x}), & \text{if } f(\boldsymbol{\theta}, \boldsymbol{x}) \leq c_2 f(\boldsymbol{\theta}^\star, \boldsymbol{x}) \\ \text{NA}, & \text{otherwise,} \end{cases} \tag{21}$$

where $\boldsymbol{\theta}^\star$ is the center of the standardized parameter space $U[0,1]^4$. The constant $c_2 > 0$ is adjusted to achieve a desired probability of missingness.

H.4 OTLcircuit function

The OTLcircuit function (Ben-Ari and Steinberg, 2007) is given as

$$f(\boldsymbol{\theta}, \boldsymbol{x}) = \frac{(V_{b1} + 0.74)\beta(R_{c2} + 9)}{\beta(R_{c2} + 9) + R_f} + \frac{11.35 R_f}{\beta(R_{c2} + 9) + R_f} + \frac{0.74 R_f \beta(R_{c2} + 9)}{(\beta(R_{c2} + 9) + R_f) R_{c1}}, \tag{22}$$

where

$$V_{b1} = \frac{12 R_{b2}}{R_{b1} + R_{b2}},$$

and the variable partition $(\boldsymbol{\theta}, \boldsymbol{x})$ and the respective ranges are shown in Table 6.

	Variable	Range
$\boldsymbol{\theta}$	R_f	[0.5, 3]
	β	[50, 300]
$\boldsymbol{x}$	R_{b1}	[50, 150]
	R_{b2}	[25, 70]
	R_{c1}	[1.2, 2.5]
	R_{c2}	[0.25, 1.2]

Table 6: Variable ranges for the OTLcircuit function in (22).

The function produces a MNAR that follows the"logistic missing not-at-random" scheme used in Muzellec et al. (2020), where the probability of missing is according to a logistic model over the columns.

I Full results for surrogate comparison experiment

This section presents the full simulation results following the numerical experiments detailed in the main article. The surrogate methods compared are the proposed method PCGPwM, two simplistic imputation methods PCGP-kNN and PCGP-BR, EMGP, colGP, and GP-OM. The surrogate methods are tested under nine missingness scenarios: (MCAR, 1%), (MCAR, 5%), (MCAR, 25%), (MNAR, 1%), (MNAR, 5%), (MNAR, 25%), (MAR, 1%), (MAR, 5%), and (MAR, 25%), where results for the scenario (MNAR, 5%) is reported in the main text of the article. To summarize the metrics of surrogate quality, the test RMSE is recorded to measure the accuracy of the surrogate; the 90% coverage and the 90% width are used to assess the uncertainty quantification of the surrogate.

I.1 Results under MNAR

The results under MNAR are presented, in the order of 1%, 5%, and 25%.

Figure 5 shows the test RMSEs for the surrogate methods under 1% missingness. The simplistic imputation methods fail to reduce the errors (i.e., fail to improve surrogate predictions) as N grows

for two of the functions (borehole and wingweight). The corresponding missingness mechanism is MNAR, depending on the unobserved function value. The simplistic imputation methods reduce the errors in the other two functions (piston and OTLcircuit). The RMSEs from PCGPwM and the other methods decrease as N increases in all four functions. The EMGP method consistently results in larger error than PCGPwM, and colGP. The colGP method performs well across functions. However, colGP will not be able to compete in terms of its computation time at higher output dimensions.

Figure 6 shows the 90% coverage under 1% missingness, and Figure 7 shows the 90% width. GP-OM results in a high coverage in the borehole function but poor coverages in the other functions. The 90% widths show that GP-OM results in a confidence interval too wide in the borehole and too narrow in the others. The EMGP method achieves the prescribed coverage in the piston and OTLcircuit functions, while undercovering in the borehole function. The coverage from EMGP improves as N grows and achieves the prescribed level for larger Ns. PCGPwM, the simplistic imputation methods, and colGP result in adequate coverages in all functions. However, the simplistic imputation methods maintain wider intervals than PCGPwM in all four cases.

Figure 8 shows the test RMSEs under 25% MNAR. Similar conclusions to the 1% case can be drawn about the simplistic imputation methods, where they fail to improve in two functions given more data. The other methods exhibit similar behaviors in predictive accuracy.

Figure 9 shows the coverages under 25% MNAR and Figure 10 shows the widths of the intervals. We observe that GP-OM and the simplistic imputation methods result in similar behavior as the other cases, meaning that GP-OM results in high coverage in only one function, and the simplistic imputation method achieves high coverage with generally wide intervals. PCGPwM results in adequate coverages with narrower intervals for all functions except the OTLcircuit function for $n = 2500$.

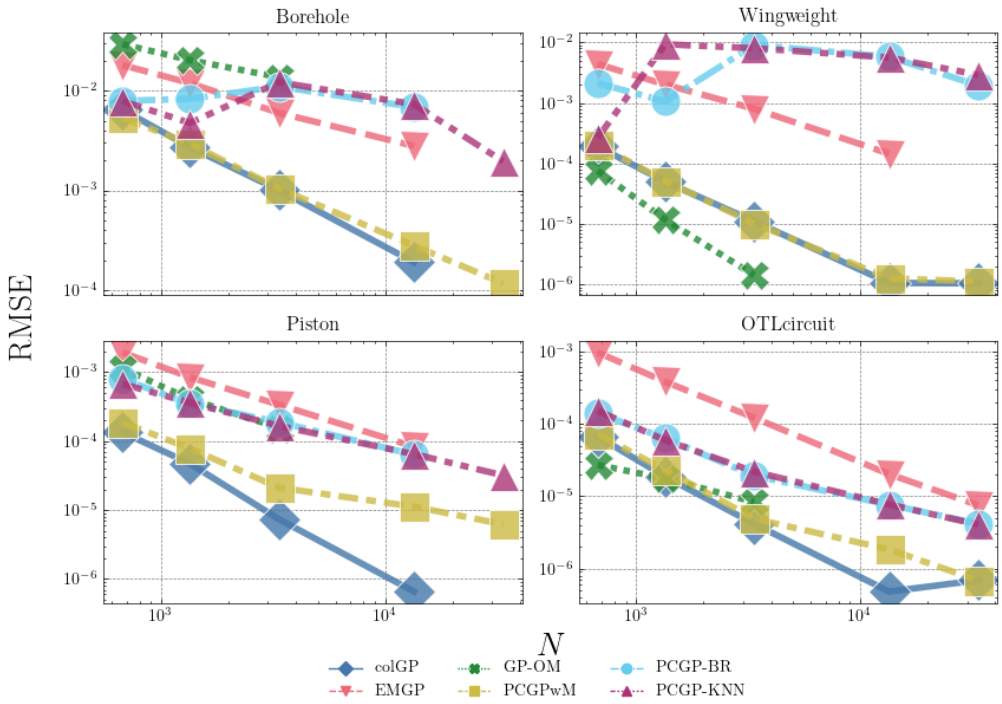

Figure 5: Comparison of prediction accuracy of surrogate methods, under 1% MNAR.

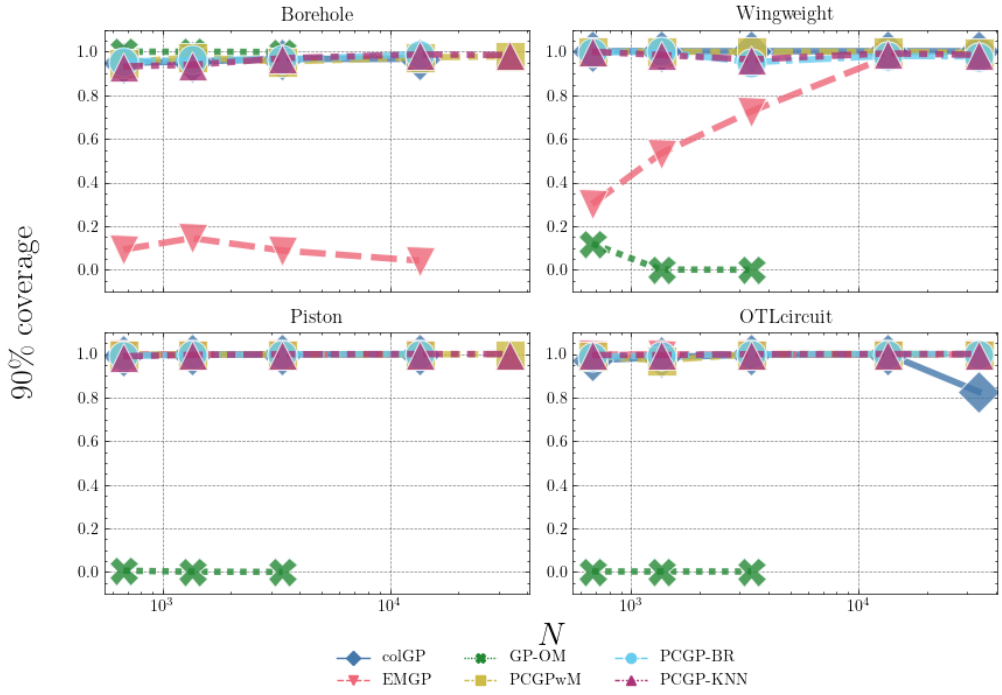

Figure 6: Comparison of 90% coverage of surrogate methods, under 1% MNAR.

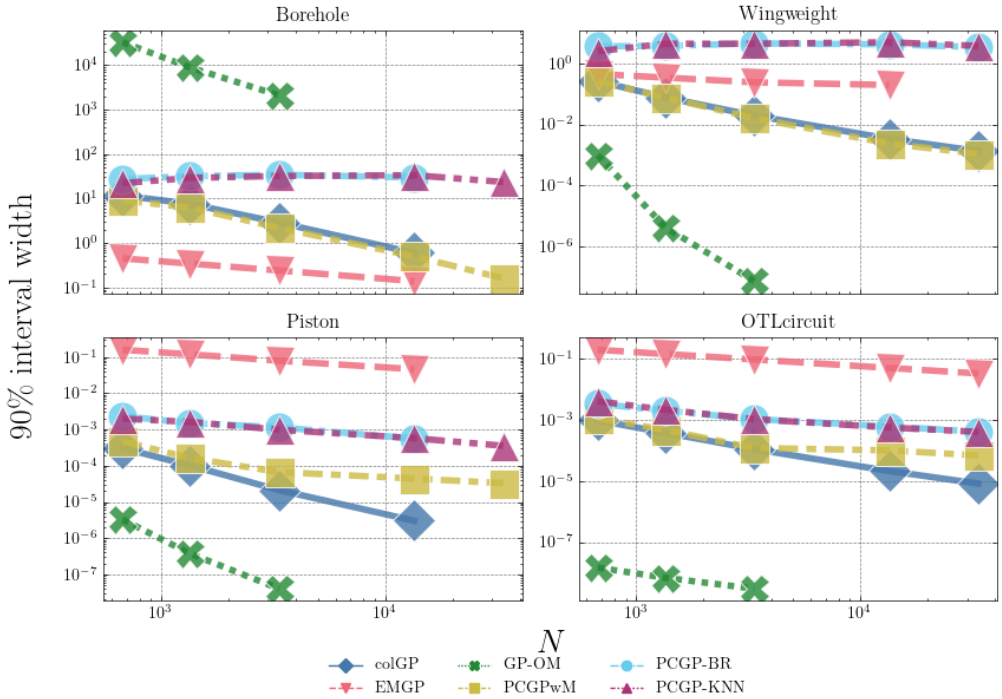

Figure 7: Comparison of 90% width of surrogate methods, under 1% MNAR.

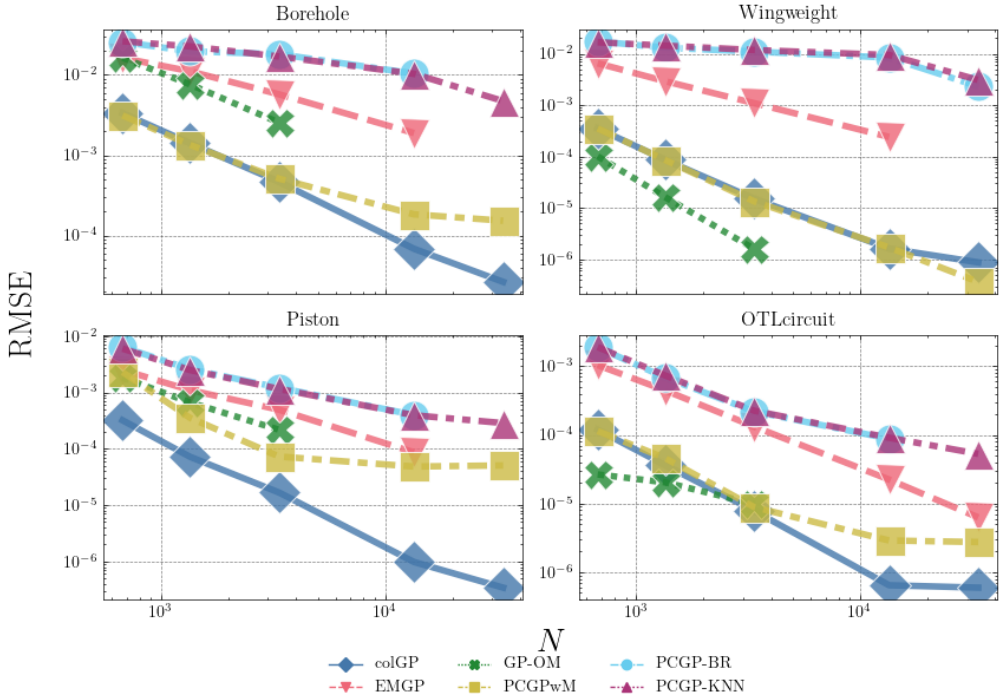

Figure 8: Comparison of prediction accuracy of surrogate methods, under 25% MNAR.

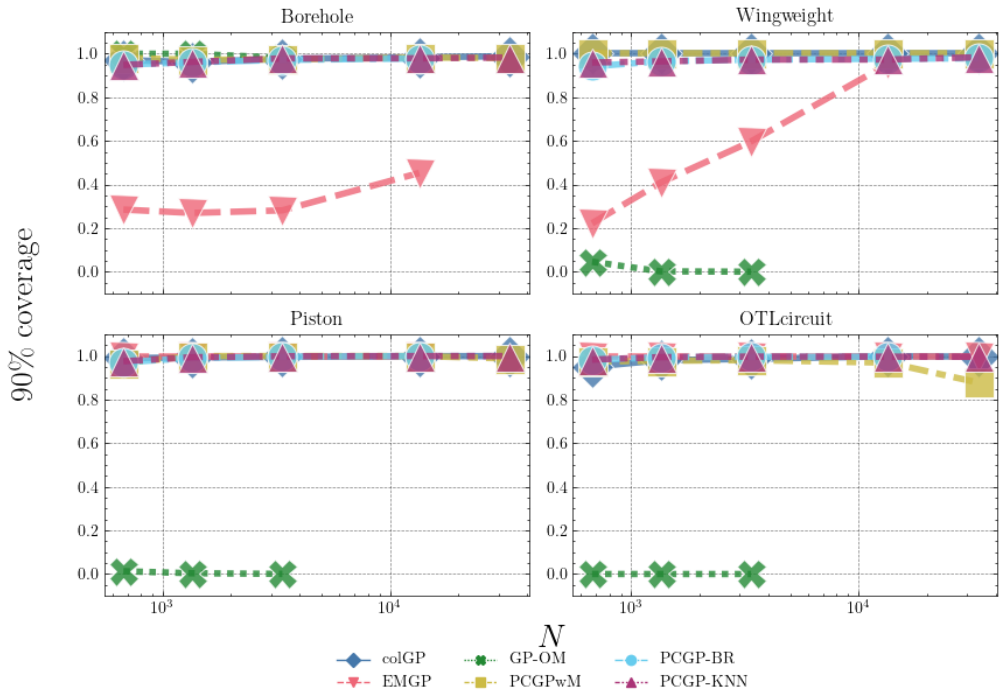

Figure 9: Comparison of 90% coverage of surrogate methods, under 25% MNAR.

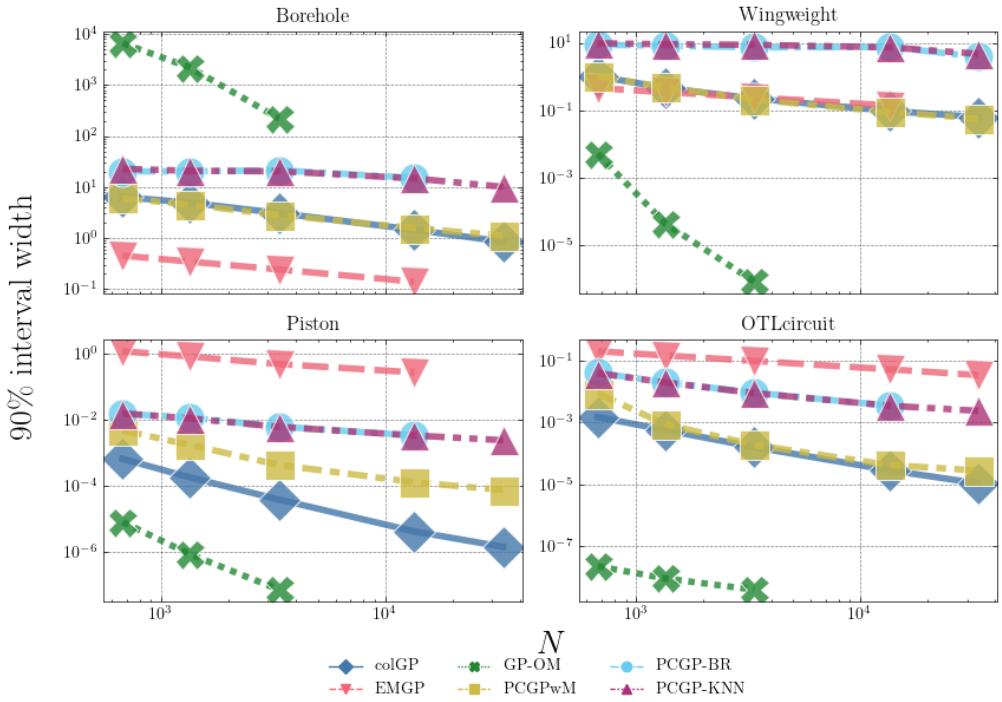

Figure 10: Comparison of 90% width of surrogate methods, under 25% MNAR.

I.2 Results under MCAR

The results under MCAR are presented, in the order of 1%, 5%, and 25%.

Figure 11 shows the test RMSEs for the surrogate methods under 1% MCAR. The RMSEs decrease as N increases for all methods in all four test functions. Among the principal component-based methods, PCGPwM generally achieves lower errors than the simplistic imputation methods. The colGP method performs well across the functions, and the EMGP method generally results in higher error compared to the other methods.

Figure 12 shows the 90% coverages, and Figure 13 shows the corresponding 90% width of the intervals. Similar to the MNAR case, GP-OM produces a high coverage in the borehole function, but near-zero coverages in the others. The corresponding 90% widths reflect the high coverage with wide intervals and the undercoverages with intervals that are too narrow. The EMGP method achieves adequate coverage for the piston and OTLcircuit functions, but undercovers in the borehole function. The coverage for the wingweight function improves as N increases and achieves the prescribed level at larger Ns. All the other methods achieve adequate coverages, where the corresponding widths decrease as N increases. PCGPwM generally produces narrower widths than do the simplistic imputation methods while attaining comparable coverages.

Figure 14 shows the test RMSEs under 5% MCAR. Figures 15 and 16 show the 90% coverages and the 90% widths, respectively. The results of the surrogate accuracy with 1% MCAR extends to the scenario with 5% MCAR, except the levels of coverage for PCGPwM has decreased in large N. The 90% widths decrease for all methods as N increases.

Figure 17 shows the test RMSEs under 25% MCAR. The RMSEs for all the methods generally decrease with N. Figures 18 and 19 show the 90% coverages and the 90% widths, respectively. Similar conclusions can be drawn for GP-OM, compared to lower missingness scenarios. The simplistic imputation methods maintain adequate coverages while maintaining slightly wider intervals when

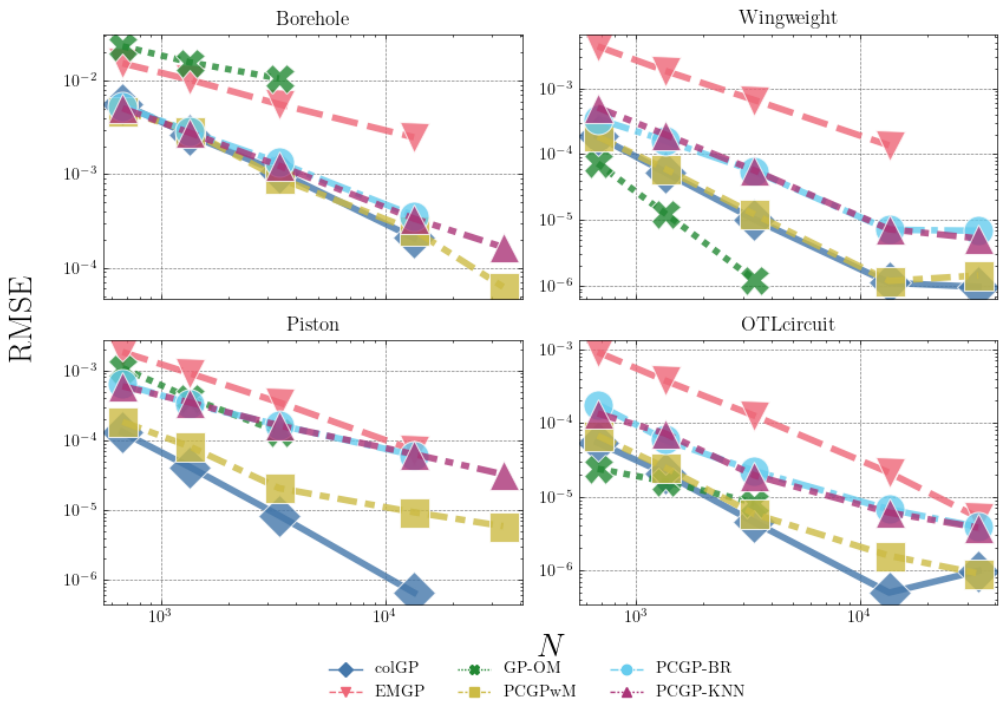

Figure 11: Comparison of prediction accuracy of surrogate methods, under 1% MCAR.

compared with PCGPwM. The EMGP method exhibits similar behavior, where coverage improves with the borehold and wingweight functions as N increases.

I.3 Results under MAR

The results under MAR are presented, in the order of 1%, 5%, and 25%. The numerical results under MAR follow similar trends as MCAR for all percentages of missingness. We refer the analysis to the reporting of results in the MCAR section. Figures 20, 23, and 26 present the corresponding RMSEs. Figures 21, 24, and 27 present the 90% coverages. And Figures 22, 25, and 28 present the 90% widths.

I.4 Comparison of PCGPwM against a baseline method

We compare PCGPwM against a baseline method where we build the principal components as if the complete data are provided (i.e., $\mathbf{\Phi}$ and $\mathbf{\Lambda}$). We label it the "PCGP-benchmark" method. The

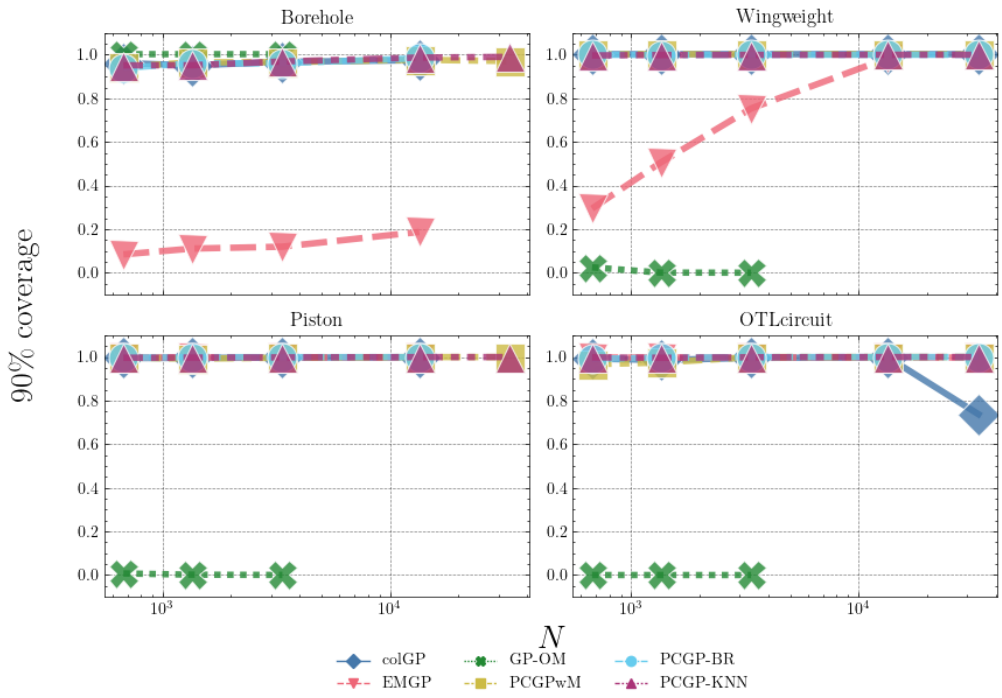

Figure 12: Comparison of 90% coverage of surrogate methods, under 1% MCAR.

purpose of including this method is to examine the significance of the estimation of the principal components from missing data. Although improved estimation of principal components from missing data is outside the scope of this article, by studying this method we can offer insights into the limitations of the proposed method.

The two methods are compared under the 5% MNAR scenario. The results in terms of RMSE, 90% coverage, and the 90% interval width show that PCGPwM performs near identically to PCGP-benchmark, thus the figures are left out. In other words, the proposed imputation recovers the unknown principal components well. In certain cases, when compared with colGP (which uses the data without dimension reduction), both PCGPwM and PCGP-benchmark cannot further reduce error even with larger data size. This points to a potential limitation in the use of principal component methods.

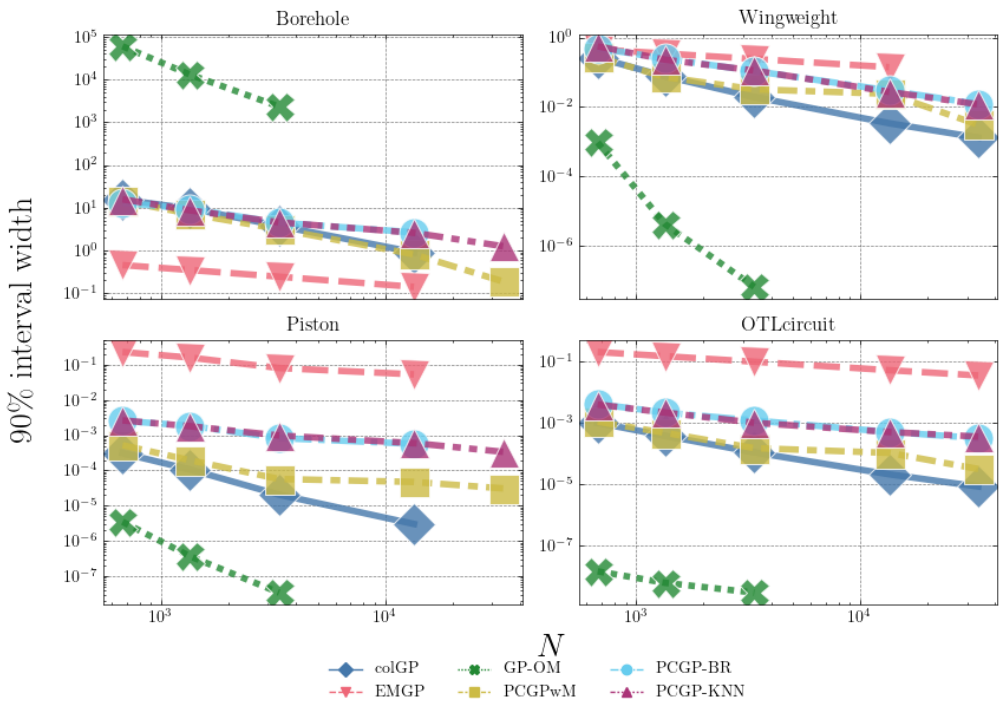

Figure 13: Comparison of 90% width of surrogate methods, under 1% MCAR.

J Scaling of Fayans EDF parameter space

Table 7 contains the unscaled centroid, the length scales, and the lower and upper bounds for each dimension of the parameter. This information is reproduced from Table 5 of Bollapragada et al. (2021).

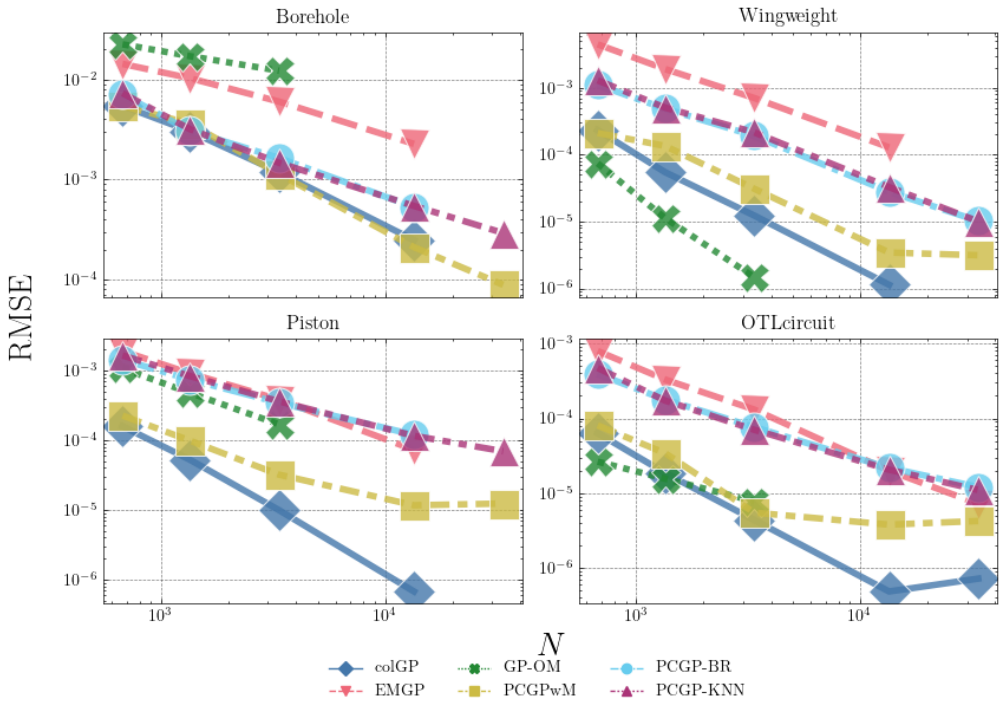

Figure 14: Comparison of prediction accuracy of surrogate methods, under 5% MCAR.

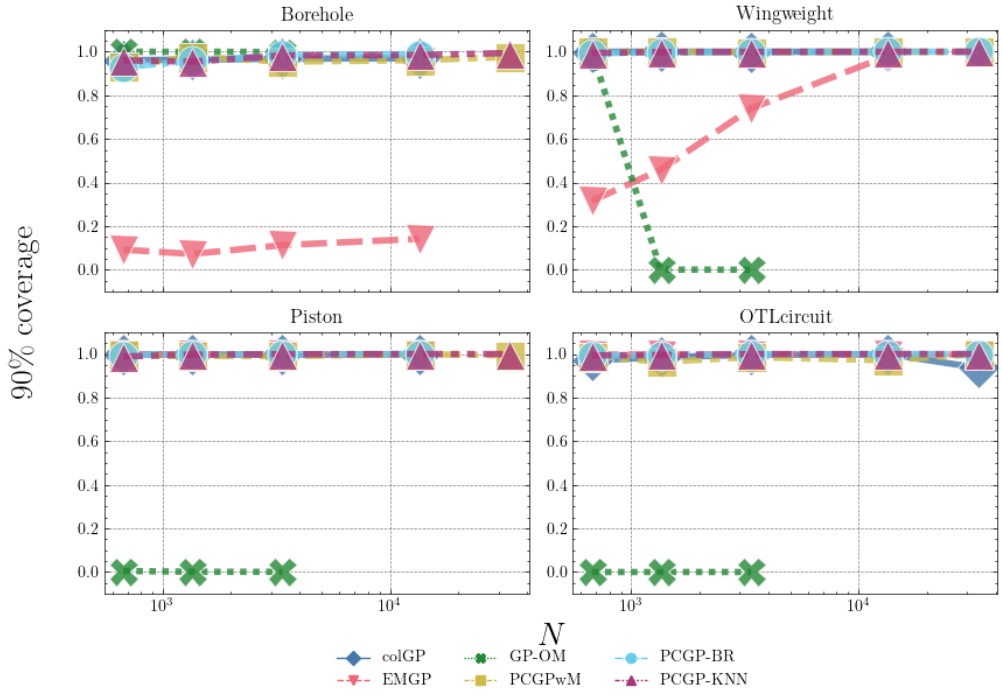

Figure 15: Comparison of 90% coverage of surrogate methods, under 5% MCAR.

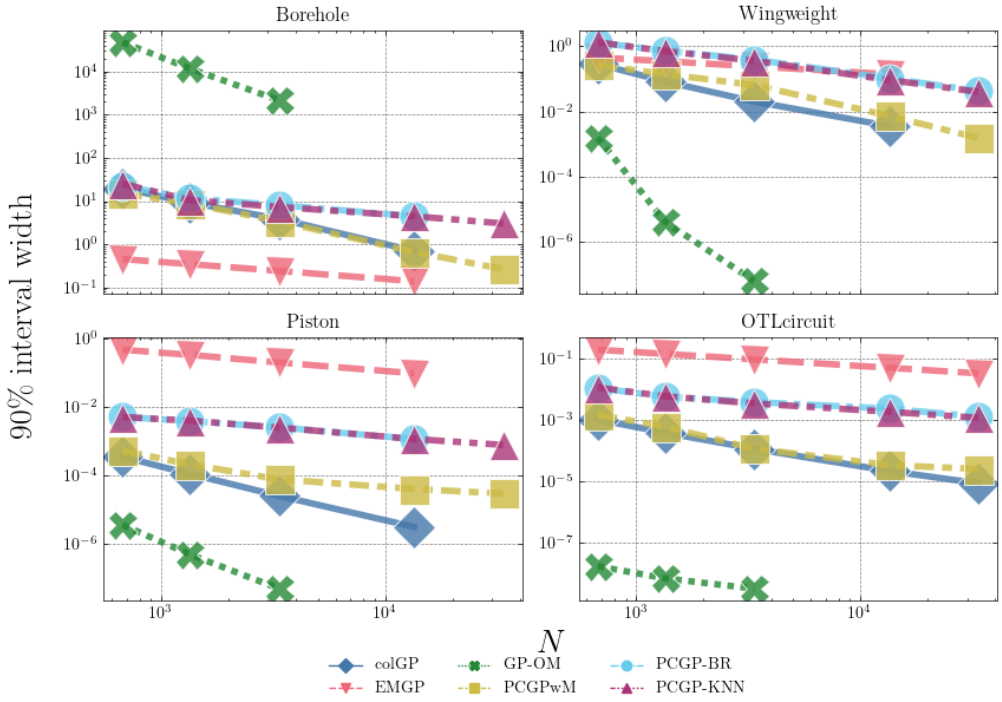

Figure 16: Comparison of 90% width of surrogate methods, under 5% MCAR.

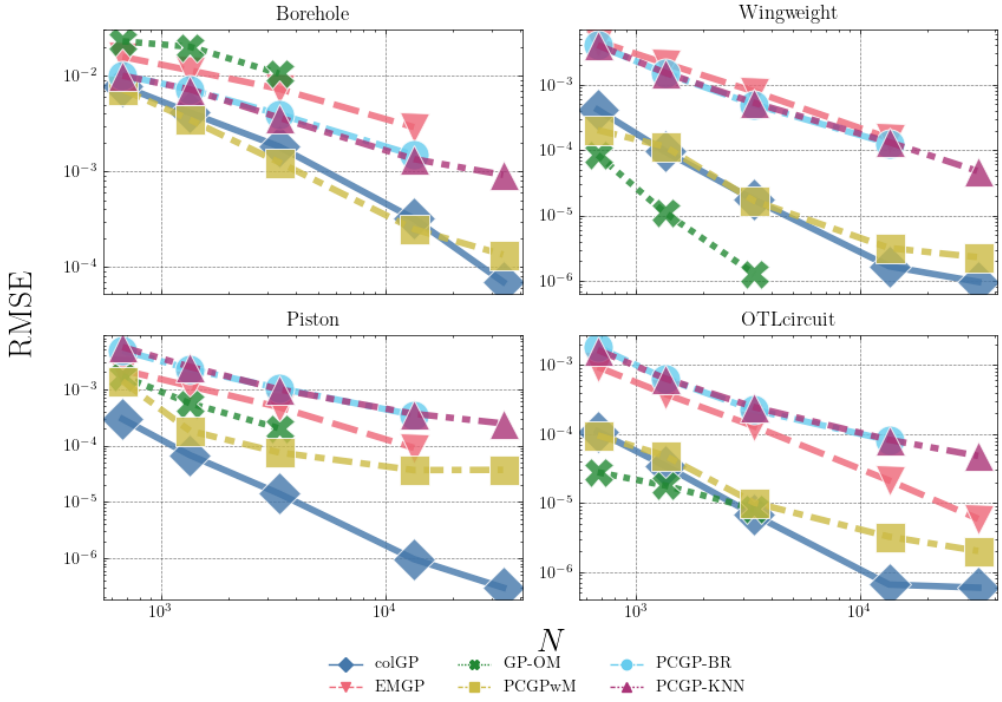

Figure 17: Comparison of prediction accuracy of surrogate methods, under 25% MCAR.

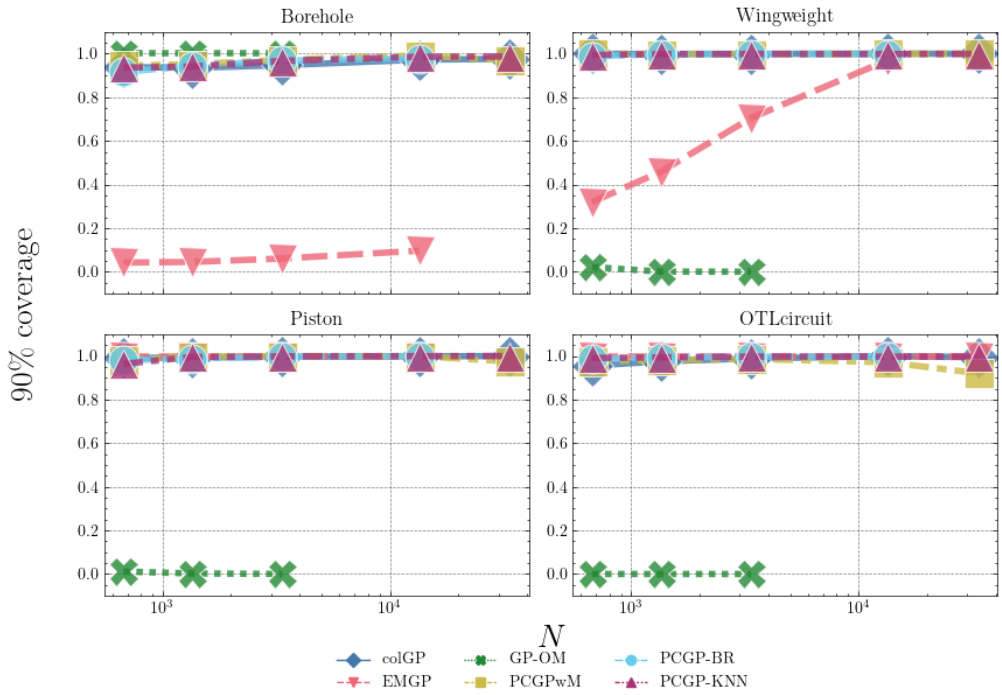

Figure 18: Comparison of 90% coverage of surrogate methods, under 25% MCAR.

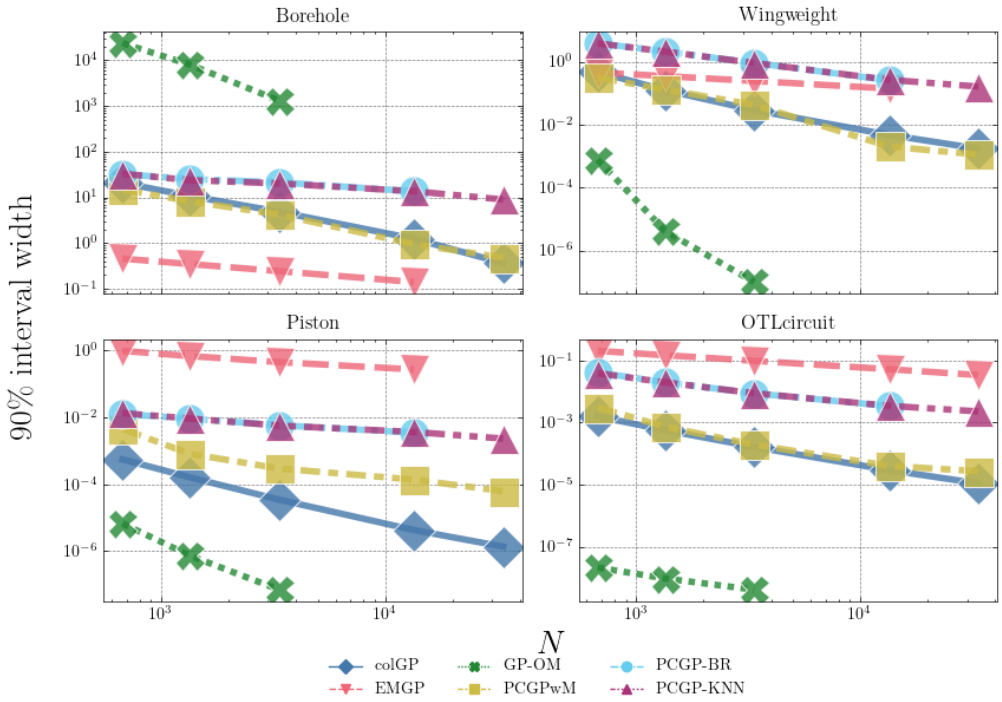

Figure 19: Comparison of 90% width of surrogate methods, under 25% MCAR.

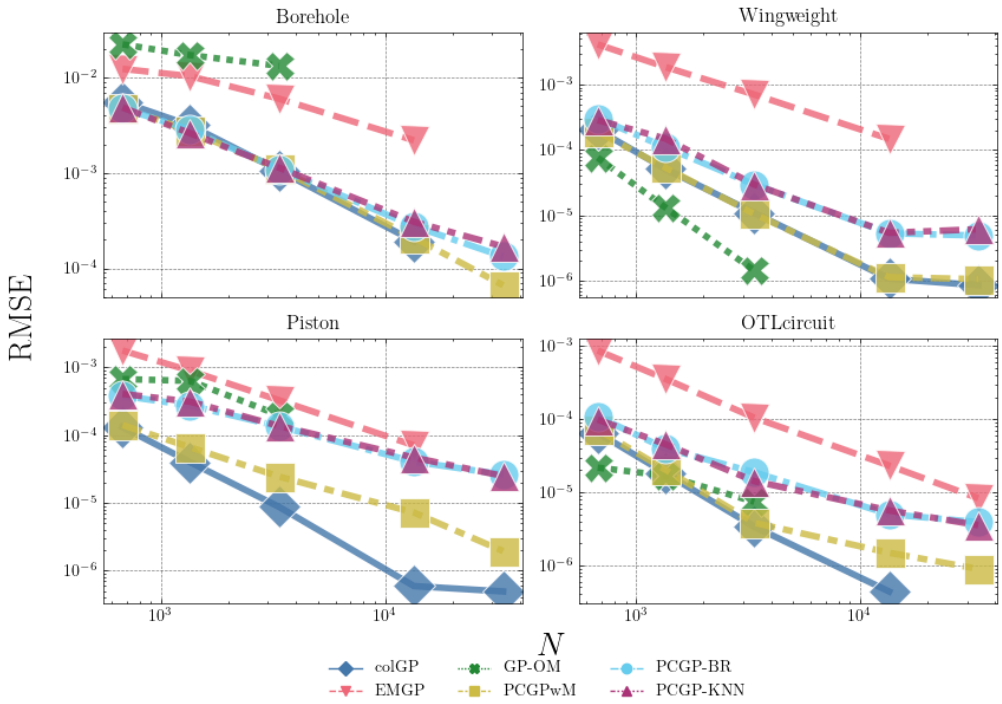

Figure 20: Comparison of prediction accuracy of surrogate methods, under 1% MAR.

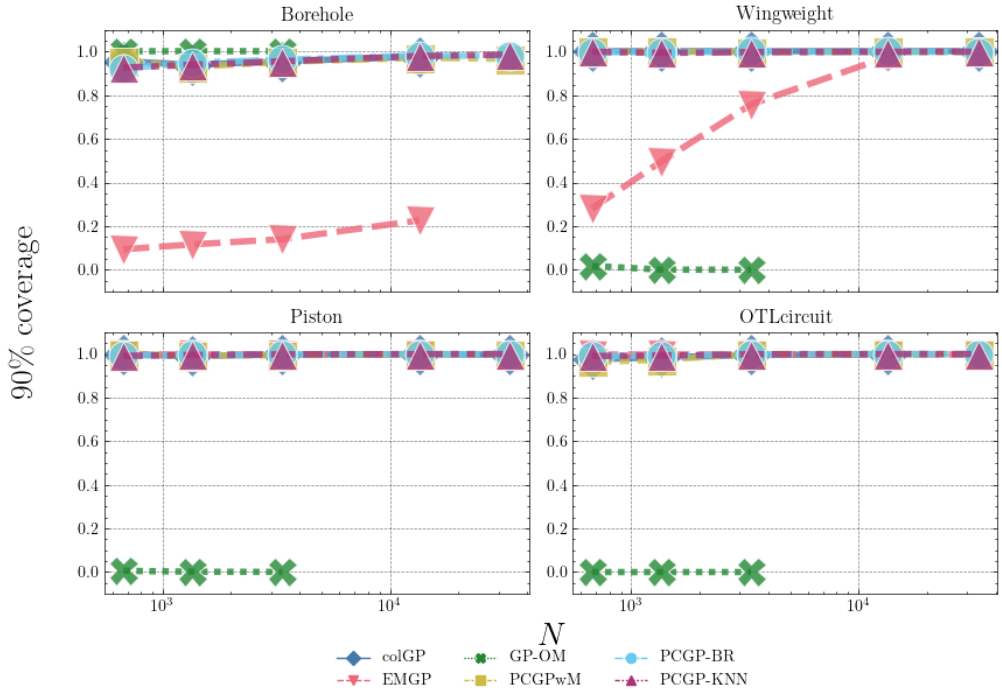

Figure 21: Comparison of 90% coverage of surrogate methods, under 1% MAR.

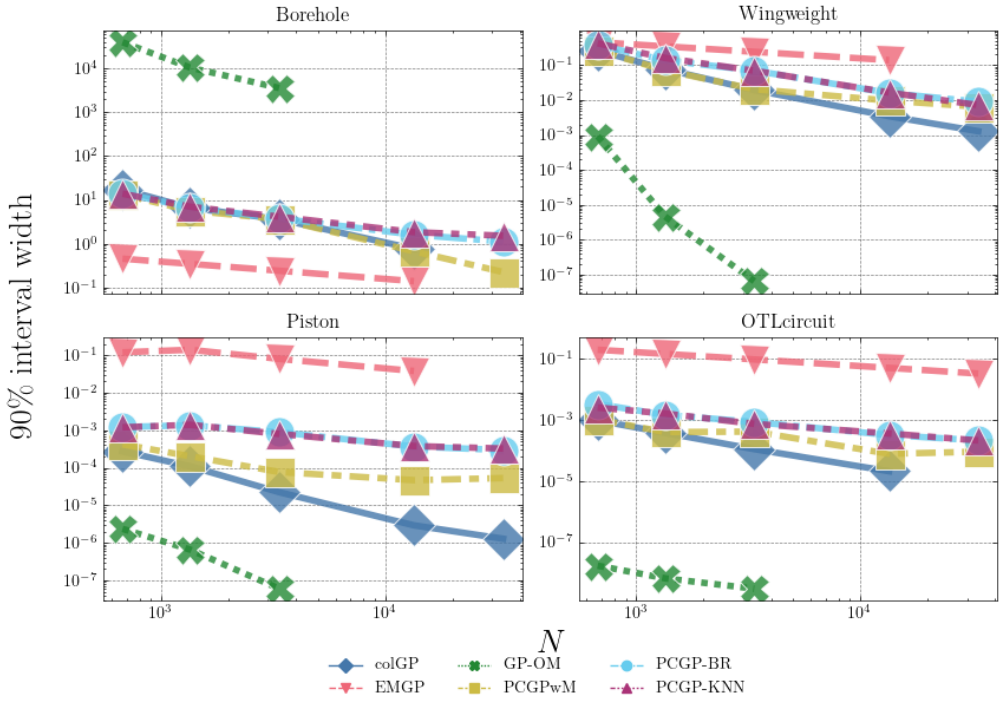

Figure 22: Comparison of 90% width of surrogate methods, under 1% MAR.

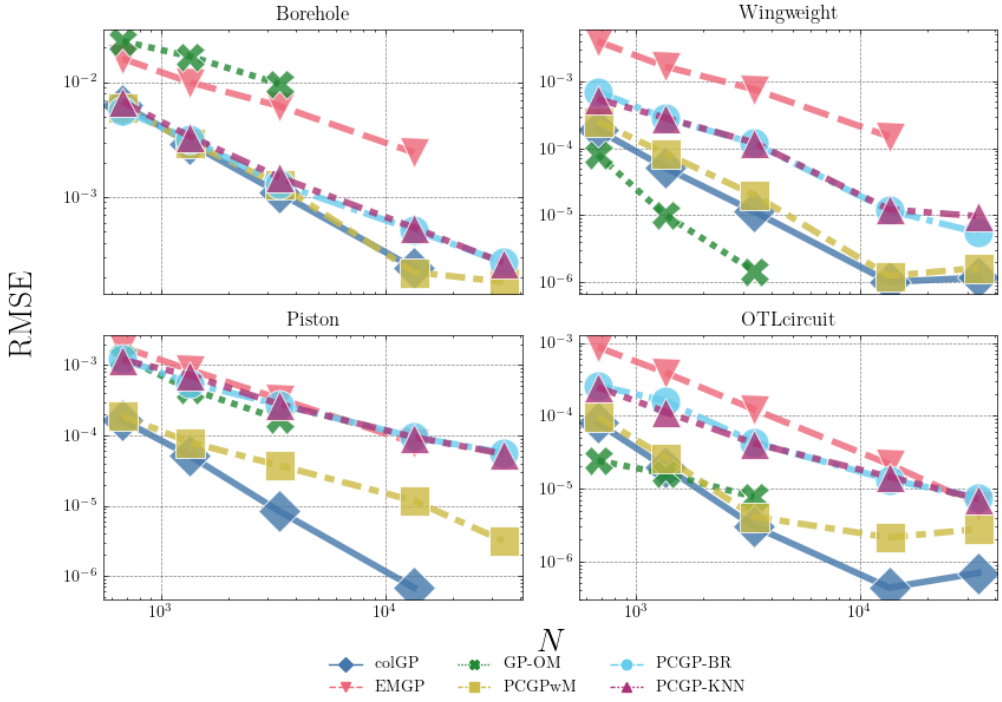

Figure 23: Comparison of prediction accuracy of surrogate methods, under 5% MAR.

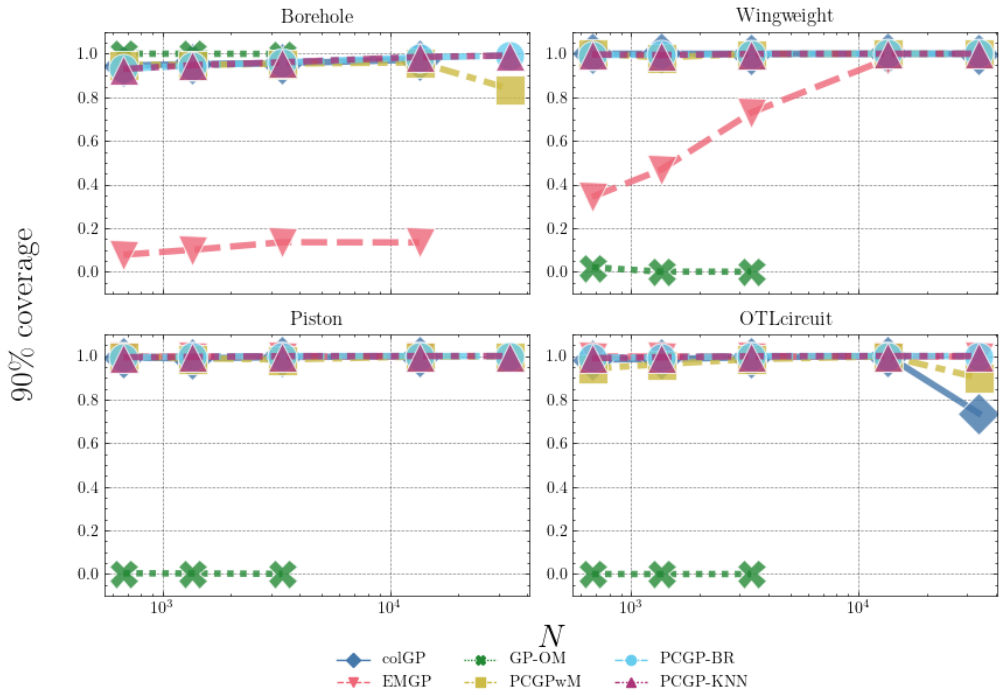

Figure 24: Comparison of 90% coverage of surrogate methods, under 5% MAR.

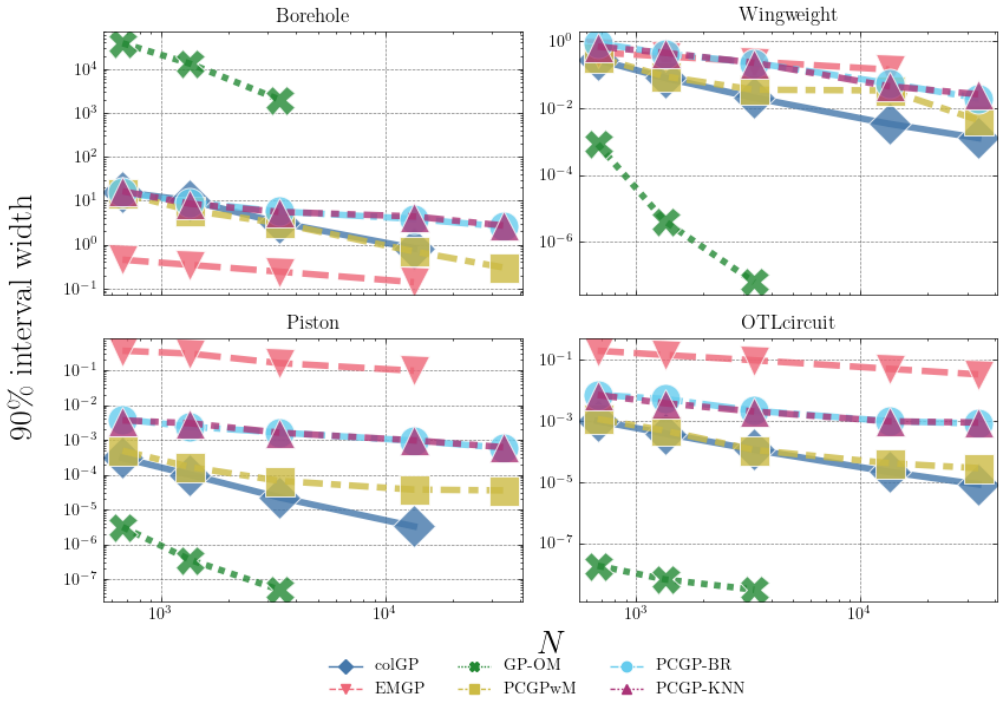

Figure 25: Comparison of 90% width of surrogate methods, under 5% MAR.

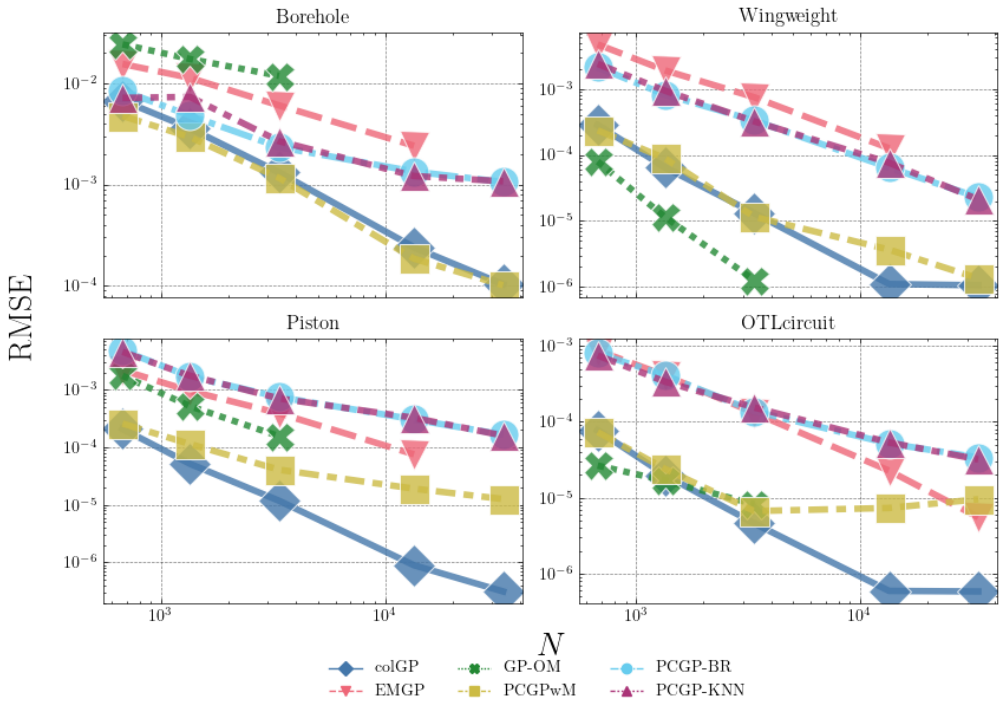

Figure 26: Comparison of prediction accuracy of surrogate methods, under 25% MAR.

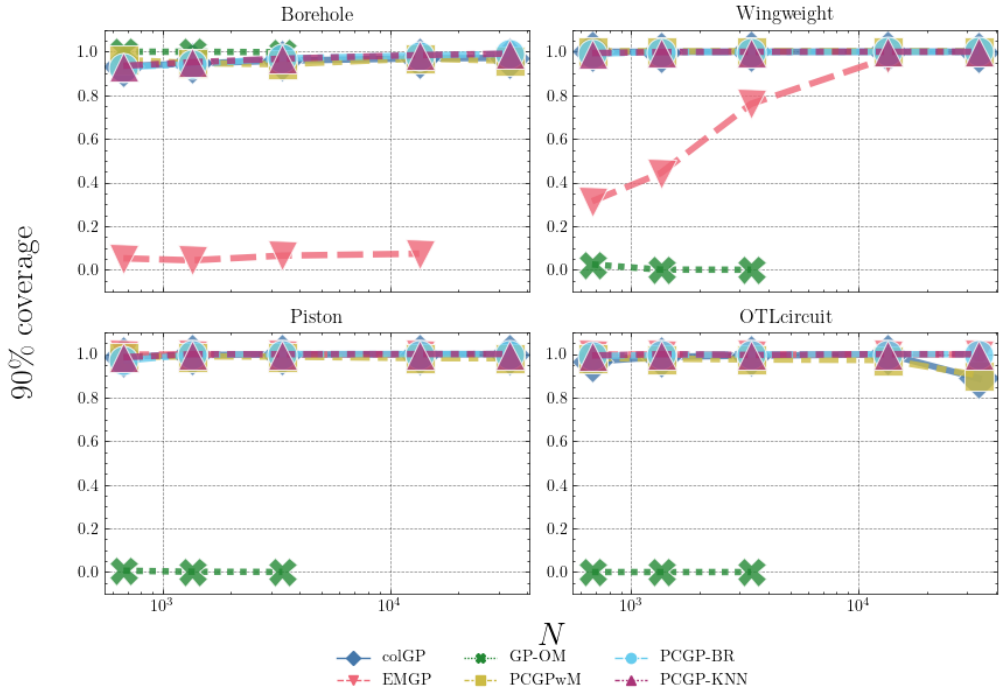

Figure 27: Comparison of 90% coverage of surrogate methods, under 25% MAR.

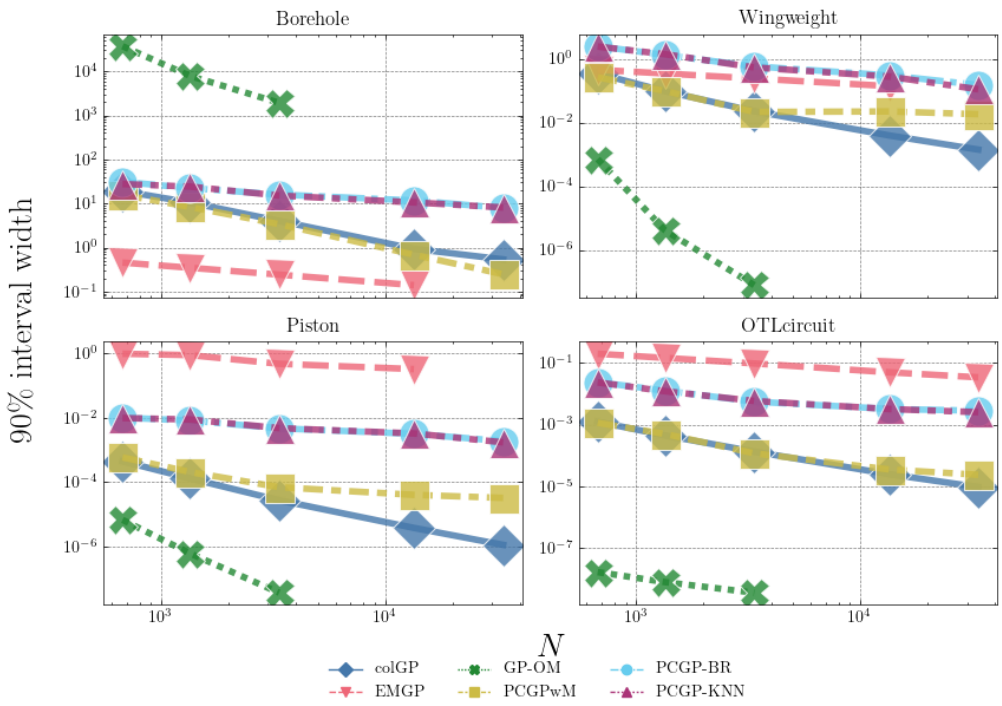

Figure 28: Comparison of 90% width of surrogate methods, under 25% MAR.

Parameter	unscaled center	scale	lb	ub
ρ_{eq}	0.1642	0.004	0.146	0.167
E/A	−15.86	0.1	−16.21	−15.50
K	206.6	25	137.2	234.4
J	28.3	3.2	19.5	37.0
L	35.9	32	2.2	69.6
h^{v}_{2-}	11.34	19.01	0	100
a^{s}_{+}	0.562	0.06	0.418	0.706
h^{s}_{∇}	0.460	0.24	0	0.516
κ	0.188	0.02	0.076	0.216
κ'	0.045	0.17	−0.892	0.982
f^{ξ}_{ex}	−4.46	1.16	−4.62	−4.38
h^{ξ}_{∇}	4.18	1.68	3.94	4.27
h^{ξ}_{+}	3.44	1.4	−0.96	3.66

Table 7: Fayans EDF model parameters and their scaling information, reproduced from Table 5 of Bollapragada et al. (2021).