Residual Gaussian process: A tractable nonparametric Bayesian emulator for multi-fidelity simulations

W.W. Xing [b,c], A.A. Shah [a,*], P. Wang [b,e], S. Zhe [d], Q. Fu [a], R.M. Kirby [c]

[a] *School of Energy and Power Engineering, Chongqing University, 174 Shazhengjie, Shapingba, Chongqing 400044, China*
[b] *School of Integrated Circuit Science and Engineering, Beijing University of Aeronautics and Astronautics, Address: No. 37 Xueyuan Road, Haidian District 100191, China*
[c] *Scientific Computing and Imaging Institute, University of Utah, 72 S Central Campus Drive, Salt Lake City, UT 84112, United States*
[d] *School of Computing, University of Utah, 72 S Central Campus Drive, Room 3750, Salt Lake City, UT 84112, United States*
[e] *LMIB & School of Mathematical Sciences, Beihang University, Beijing, China*

A R T I C L E   I N F O

A B S T R A C T

Challenges in multi-fidelity modelling relate to accuracy, uncertainty estimation and high-dimensionality. A novel additive structure is introduced in which the highest fidelity solution is written as a sum of the lowest fidelity solution and residuals between the solutions at successive fidelity levels, with Gaussian process priors placed over the low fidelity solution and each of the residuals. The resulting model is equipped with a closed-form solution for the predictive posterior, making it applicable to advanced, high-dimensional tasks that require uncertainty estimation. Its advantages are demonstrated on univariate benchmarks and on three challenging multivariate problems. It is shown how active learning can be used to enhance the model, especially with a limited computational budget. Furthermore, error bounds are derived for the mean prediction in the univariate case.

1. Introduction

The design, optimization, and control of many systems in science and engineering can rely heavily on computational modelling. Different approaches can be adopted, depending upon the problem at hand or the computational budget available. One way to categorize a computational model is via its *fidelity*. Roughly speaking, low-fidelity computational models are those of low complexity or resolution in terms of the physics or any adjustable setting of the computer-based approximation [1]. High-fidelity models are defined in a similar manner. The former are usually associated with a lower computational burden, the penalty for which is a loss in accuracy. In all computational models, the settings can be adjusted to obtain outputs of different fidelities (the grid point spacing, time step, order of an approximating basis, error tolerances, and so on). It is also usually the case that there is a range of mathematical models to describe a given physical problem, with different levels of physical detail incorporated into the equations, initial-boundary conditions and geometry that make up the model.

In some applications, high-fidelity models are impractical, especially when a high number of runs of the model is required across a parameter (input) space. In such applications, it is common to replace the model with a computationally-inexpensive approximation, termed a surrogate model [2]. Example applications include design optimization [3], real-time control [4], sensitivity analysis [5] and uncertainty quantification[6]. Surrogate models mainly rely on machine learning

---

* Corresponding author.
E-mail address: akeelshah@cqu.edu.cn (A.A. Shah).

[2,7–9] or model order reduction (MOR) [10]. MOR projects numerical formulations onto a low-dimensional subspace of the original space in which solutions are sought, and does not extend naturally to nonlinear or parameter-dependent problem. Both MOR and machine-learning methods require a large set of data generated from the original high-fidelity model (either for training or for constructing a basis), which may not be desirable or even feasible.

Another route for reducing the computational burden is multi-fidelity modelling, in which models of different fidelity are combined [2]. In most cases, multi-fidelity models involve the construction of one or more surrogate models that use information from the underlying models of different fidelity [11]. Other approaches include corrections to the low-fidelity results by leveraging information from a limited number of high-fidelity simulations, using, e.g., a Taylor series expansion or a GP model [12]. Many of the multi-fidelity methods that have been developed are specific to certain tasks, especially optimization and uncertainty quantification. Importantly, the vast majority are concerned with scalar outputs or outputs in a low-dimensional space (we refer to [1] for a recent review).

In the seminal autoregressive (AR) model of Kennedy and O'Hagan [2] for univariate outputs, a linear relationship between the different fidelity levels was assumed to hold. Le Gratiet [13] enhanced this method by employing a deterministic parametric form of the mapping from low- to high-fidelity, together with an efficient numerical scheme to reduce the computational cost. Despite its advantages, this parametric approach requires expert knowledge for model selection, as well as a large training data set. Perdikaris et al. [14] introduced the nonlinear autoregressive model (NARGP) to overcome some of these limitations. The authors placed a GP prior over the unknown cross-fidelity mapping, thereby increasing model flexibility and alleviating any overfitting issues. NARGP has been applied to a number of low-dimensional problems [14], and has been generalized to high-dimensional outputs by Parussini et al. [15]. These methods, on the other hand, lack a systematic approach to model training and rely on ad-hoc methods to select a basis for outputs in high-dimensional spaces. NARGP uses the low-fidelity solution as an input for the high-fidelity GP model, which leads to a concatenating GP structure known as the *deep GP* [16]. As a consequence, tractability is lost and expensive sampling or variational methods are required for training and inference.

Another prominent multi-fidelity approach uses stochastic collocation (SC). Narayan et al. [17] developed a greedy procedure to select low-fidelity samples and identify inputs at which to conduct a low number of high-fidelity simulations. The authors then used the low-fidelity results to approximate the coefficients for a high-fidelity SC approximation. This and related SC approaches are based on the naive assumption that adjacent fidelities share the same correlation structure, which is not always justified for complex models. The most serious drawback of SC approaches is that they require out-of-sample executions of the low-fidelity model for making predictions. When, as is frequently the case, the low-fidelity simulations are also expensive, SC approaches are not suitable for applications requiring many runs, especially sensitivity/uncertainty analyses and optimization.

NARGP and deep GP approaches becomes impractical or non-viable for multi-fidelity simulations involving outputs in high-dimensional spaces; the number of parameters in these approaches scales linearly and quadratically with the output space dimension, respectively. The recently developed Greedy NAR [18] attempts to bridge the gap between NARGP and SC, leveraging the advantages of both methods. This is achieved by a generalised AR model, in which the high-fidelity solution is given as linear map of the low-fidelity solution in a feature space. The feature map is implicitly defined by integrating out a weight matrix and kernelizing. Greedy NAR can take advantage of a sequential active learning framework to select low- and high-fidelity samples efficiently. It was shown to be more efficient and accurate than NARGP, and more flexible than SC in terms of making high-fidelity predictions since it does not rely on out-of-sample low-fidelity experiments.

In this work, an additive GP structure for multi-fidelity modelling is proposed, in which the highest fidelity solution is treated as the sum of the lowest fidelity solution and residuals between successive fidelities, over each of which a GP prior is placed. This structure leads to a flexible, tractable and highly-scalable multi-fidelity GP model, termed ResGP. Both the likelihood (for model training) and the posterior (for predictions) are given explicitly, so that expensive approximate inference methods are avoided, and no additional data is required to make predictions. Importantly, ResGPcan be scaled to high-dimensional problems, which are common candidates for surrogate modeling, without any compromise in the prediction accuracy.

Equipped with the tractable posterior, an active learning method is implemented so that ResGPcan automatically select inputs that maximize the information gain (or any desired gain) without the need for a special experimental design or *a priori* assumptions about the physical model. For univariate ResGPwe develop error bounds by assuming that the underlying high-fidelity solution is a sample from its GP prior [19], together with additional mild assumptions related to the regularities of the solution and kernel. The computational complexity of ResGPis compared with other state-of the-art methods, highlighting its advantages in terms of scalability and parameter count.

ResGPis applied to five synthetic univariate examples and three challenging multivariate examples that involve several quantities of interest. In the univariate examples ResGPis demonstrated to outperform AR, NARGP and the multi-fidelity deep GP (MF-DGP) model of Cutajar et al. [20] in terms of both prediction accuracy and uncertainty estimation in almost all cases. Compared to the other methods, the root mean square error on 1000 test points is at least 21%, 28%, 58%, 58% and 7% lower on the five examples, while the mean negative log likelihood is at least 13%, 79%, 38% and 6% lower on four of the examples. The results from the multivariate examples are compared to those from NARGP, SC and Greedy NAR, demonstrating considerable improvements in prediction accuracy, as well as stable performance for high-dimensional data sets. The normalised root mean square errors on the test sets are shown to be lower than those for the other methods

(by up to 97%) for 7 out of 9 quantities of interest across the three examples. Significantly, ResGP, especially with active learning, performs particularly well for low numbers of high-fidelity training points.

2. Statement of the problem

We are interested in numerical solutions to systems of ordinary or partial differential equations obtained from a computational model, and where repeated runs of the computational model for different input parameter values (associated with the system of equations and/or the accompanying initial-boundary conditions) are required. Such systems of equations are usually derived from conservation laws and, together with initial-boundary conditions, govern quantities of interest such as a species concentration or the temperature of a medium. Depending upon the types of equations, we may denote one such quantity of interest by $u(\mathbf{x}, t; \boldsymbol{\xi})$, $u(\mathbf{x}; \boldsymbol{\xi})$ or $u(t; \boldsymbol{\xi})$, where $\mathbf{x} \in \Omega \subset \mathbb{R}^p$, $p = 1, 2, 3$, is the spatial coordinate and t is time. $\boldsymbol{\xi} \in \mathcal{X} \subset \mathbb{R}^l$ is a vector of parameters that appear in the system of equations and/or in the initial-boundary conditions. $\mathcal{X}$ is the admissible input space, which is assumed to be a compact subset of $\mathbb{R}^l$.

Numerical solutions of the system of equations are obtained from a discretization of the equations, initial-boundary conditions and spatio-temporal domain, typically based on the finite element, finite volume, or finite difference method, together with a time-stepping scheme for the transient case. In the case of spatially-uniform systems, only a time discretization is required. The numerical solution takes the form of one or more quantities of interest at a predefined number of points in a discrete spatio-temporal grid, $\mathbf{x}_j$, $j = 1, \dots, N_x$, t_i, $i = 1, \dots, N_t$; in the case of a finite element formulation, the coefficients in a finite-element basis expansion of the quantity of interest are computed, from which values of the quantity at an arbitrary number of spatio-temporal grid points can be extracted. For each input parameter $\boldsymbol{\xi}$, the obtained values of the quantity of interest can be vectorized as follows

$$\mathbf{y}(\boldsymbol{\xi}) = \left(u(\mathbf{x}_1, t_1, \boldsymbol{\xi}), \dots, u(\mathbf{x}_{N_x}, t_1, \boldsymbol{\xi}), u(\mathbf{x}_1, t_2, \boldsymbol{\xi}), \dots, u(\mathbf{x}_{N_x}, t_{N_t}, \boldsymbol{\xi}) \right)^T, \tag{1}$$

or in some other manner that is the same for each $\boldsymbol{\xi}$. We may then treat $\mathbf{y}(\boldsymbol{\xi})$ or any scalar or vector quantity derived from $\mathbf{y}(\boldsymbol{\xi})$ as the final quantity of interest; that is, as a function $\mathbf{y} : \mathcal{X} \to \mathbb{R}^d$ of the inputs, for some integer $d \geq 1$ (for example, $d = N_x N_t$ as in Eq. (1)). Multiple quantities of interest can be modelled using the multi-fidelity method we develop by separately applying the method to each quantity. We therefore limit discussion to a single quantity of interest $\mathbf{y}(\boldsymbol{\xi})$ in the presentation of the method below.

To obtain a high-fidelity/accurate solution for $\mathbf{y}(\boldsymbol{\xi})$, we generally need to use a fine discretization in space and time, a high-order stencil, a high-order basis expansion, or tight iteration bounds. Lower-fidelity solutions can be obtained by relaxing these criteria or by using simpler physical models, e.g., spatial averaging, considering a 2-d slice or linearizing. In this way, we can obtain numerical solutions $\mathbf{y}^f(\boldsymbol{\xi})$ at different fidelities f. Other types of numerical outputs such as those from electronic-structure calculations or molecular dynamics simulations can also be modelled using the framework we develop. The requirement is simply a computational model with variable input parameters and options for generating different fidelity solutions by adjusting settings as described above.

In multi-fidelity modelling, we first conduct simulations at different fidelity levels $f = 1, \dots, F$ using inputs $\mathcal{X}^f = \{\boldsymbol{\xi}_n^f\}_{n=1}^{N_f} \subset \mathcal{X}$ to obtain outputs $\{\mathbf{y}_n^f\}_{n=1}^{N_f}$, where $\mathbf{y}_n^f = \mathbf{y}^f(\boldsymbol{\xi}_n^f)$. The outputs can be represented compactly as $\{\mathbf{Y}^f\}_{f=1}^F$, where the rows of $\mathbf{Y}^f \in \mathbb{R}^{N_f \times d}$ are the N_f solutions $\mathbf{y}_n^f$ at fidelity level f. In line with the common setting for multi-fidelity emulation [14,20], we assume that the training inputs for fidelity f are a subset of those for the preceding fidelity $f - 1$, i.e., $\mathcal{X}^f \subset \mathcal{X}^{f-1}$. We introduce an index notation for the extraction of subsets. Let $\mathbf{e}_f \subset \{1, \dots, N_{f-1}\}$, $f = 2, \dots, F$ be the indices that extract the inputs from $\mathcal{X}^{f-1}$ to obtain $\mathcal{X}^f$. Extraction of the rows $\mathbf{e}_f$ of $\mathbf{Y}^{f-1}$ leads to a matrix denoted $\mathbf{Y}_{\mathbf{e}_f}^{f-1} \in \mathbb{R}^{N_f \times d}$. Each row of $\mathbf{Y}_{\mathbf{e}_f}^{f-1}$ is an $f - 1$ fidelity solution that shares the same input as the corresponding row (output) in $\mathbf{Y}^f$. This allows us to later compactly write residuals by subtracting $\mathbf{Y}_{\mathbf{e}_f}^{f-1}$ from the matrix $\mathbf{Y}^f$.

The goal of this paper is to accurately approximate one or more high-fidelity quantities of interest $\mathbf{y}^F(\boldsymbol{\xi})$ by efficiently combining lower-fidelity information. We are especially interested in quantities of interest that lie in high-dimensional spaces (large d). Moreover, the method we develop will be able to evaluate the predictive uncertainty efficiently and effectively. We also develop an efficient design-of-experiment that leads to optimal surrogates of the quantities of interest given limited computational resources. Specifically, the method selects locations at which to conduct the high-fidelity experiments according to the maximum information gain. We also provide an analysis of the time and space complexity of our method and develop error bounds for the univariate case.

3. Residual Gaussian process model

In this section we introduce a novel tractable and scalable structure to model multi-fidelity simulation data. Rather than imposing a concatenating structure as in NARGP, which forgoes the tractable nature of a GP [21], we decompose the GP defined over the high-fidelity output into a sum of GPs relating to the differences between successive fidelities. Specifically,

we impose the following residual structure

$$\mathbf{y}^f(\boldsymbol{\xi}) = \sum_{k=1}^{f} \mathbf{r}^k(\boldsymbol{\xi}), \quad f = 1, \ldots, F, \tag{2}$$

where $\mathbf{r}^f(\boldsymbol{\xi}) = \mathbf{y}^f(\boldsymbol{\xi}) - \mathbf{y}^{f-1}(\boldsymbol{\xi})$, $f = 2, \ldots, F$, are the residual functions between fidelities. The function $\mathbf{r}^1(\boldsymbol{\xi})$, on the other hand, is defined as $\mathbf{r}^1(\boldsymbol{\xi}) = \mathbf{y}^1(\boldsymbol{\xi})$. Note that we present the model for the multivariate case. The univariate case can be obtained from the formulae presented below in an obvious manner.

The unknown functions $\mathbf{r}^f(\boldsymbol{\xi})$ are then treated as random processes that can be approximated using any probabilistic data-driven model. We place an independent, zero-mean GP prior over each residual function $\mathbf{r}^f(\boldsymbol{\xi})$, i.e.,

$$\mathbf{r}^f(\boldsymbol{\xi}) \sim \mathcal{GP}(\mathbf{r}^f(\boldsymbol{\xi}) \mid \mathbf{0}, k^f(\boldsymbol{\xi}, \boldsymbol{\xi}'|\boldsymbol{\theta}^f) \otimes \boldsymbol{\Omega}^f + \delta(\boldsymbol{\xi}, \boldsymbol{\xi}')\tau^f\mathbf{I}), \quad f = 1, \ldots, F, \tag{3}$$

where $\boldsymbol{\Omega}^f \in \mathbb{R}^{d \times d}$ is an unknown coregionalization matrix corresponding to the correlations between the components of $\mathbf{r}^f(\boldsymbol{\xi})$, $\otimes$ is the Kronecker product and $\delta(\boldsymbol{\xi}, \boldsymbol{\xi}')$ is the Kronecker-delta function. Without loss of generality, the mean is assumed to be identically zero by centering the observations. We note that this is the multivariate GP model of Conti and O'Hagan [7], with a separable covariance structure. That is, the covariance matrices assume the forms $\mathbf{K}^f \otimes \boldsymbol{\Omega}^f$, in which $[\mathbf{K}^f]_{nm} = k^f(\boldsymbol{\xi}_n, \boldsymbol{\xi}_m|\boldsymbol{\theta}^f)$, $\boldsymbol{\xi}_n, \boldsymbol{\xi}_m \in \mathcal{X}^f$, captures the correlations between values of $\mathbf{r}^f(\boldsymbol{\xi})$ at different inputs $\boldsymbol{\xi}$ and $\boldsymbol{\Omega}^f$ captures the spatial correlations, i.e., between different components of $\mathbf{r}^f(\boldsymbol{\xi})$. The covariance functions $k^f(\boldsymbol{\xi}, \boldsymbol{\xi}|\boldsymbol{\theta})$ contain unknown hyperparameters $\boldsymbol{\theta}^f$. The terms $\delta(\boldsymbol{\xi}, \boldsymbol{\xi}')\tau^f\mathbf{I}$ are included to account for zero-mean, i.i.d. measurement error (across inputs and spatial coordinates), or, equivalently, as regularization terms to prevent ill-conditioning during training. The τ^f are treated as hyperparameters or (optionally) fixed to some small values in the regularization interpretation [22]. The distribution (3) is conditioned on the full set of hyperparameters $\{\boldsymbol{\theta}^f, \Omega^f, \tau^f\}$ but to avoid notational clutter, conditioning on hyperparameters and inputs or observations is not explicitly indicated. In the measurement error interpretation, the targets, i.e., given values of $\mathbf{r}^f(\boldsymbol{\xi})$, are considered to be values of a latent function $\mathbf{r}_l^f(\boldsymbol{\xi})$ corrupted by the i.i.d. noise $\boldsymbol{\epsilon}^f$

$$\mathbf{r}^f(\boldsymbol{\xi}) = \mathbf{r}_l^f(\boldsymbol{\xi}) + \boldsymbol{\epsilon}^f, \quad f = 1, \ldots, F, \tag{4}$$

with priors $\mathbf{r}_l^f(\boldsymbol{\xi}) \sim \mathcal{GP}(\mathbf{r}_l^f(\boldsymbol{\xi}) \mid \mathbf{0}, k^f(\boldsymbol{\xi}, \boldsymbol{\xi}'|\boldsymbol{\theta}^f) \otimes \boldsymbol{\Omega}^f)$ and $\boldsymbol{\epsilon}^f \sim \mathcal{GP}(\boldsymbol{\epsilon}^f \mid \mathbf{0}, \delta(\boldsymbol{\xi}, \boldsymbol{\xi}')\tau^f\mathbf{I})$.

Choosing the right kernel function for a specific application is non-trivial. When there is no prior knowledge to guide the choice, the automatic relevance determinant (ARD) kernel [21]

$$k^f(\boldsymbol{\xi}, \boldsymbol{\xi}'|\boldsymbol{\theta}^f) = \theta_0^f \exp\left(-(\boldsymbol{\xi} - \boldsymbol{\xi}')\mathrm{diag}(\theta_1^f, \ldots, \theta_1^f)(\boldsymbol{\xi} - \boldsymbol{\xi}')^T\right), \quad f = 1, \ldots, F, \tag{5}$$

with $\boldsymbol{\theta}^f = (\theta_0^f, \ldots, \theta_1^f)^T$ is often used. The ARD kernel can freely capture the influence of each individual input (coordinate of $\boldsymbol{\xi}$) on the output. The hyperparameters $\{\tau^f, \boldsymbol{\theta}^f\}$ can be estimated by maximizing the log-marginal likelihood (see Section 3.1).

The high-fidelity prior in this model can be written as follows

$$\mathbf{y}^F(\boldsymbol{\xi}) \sim \mathcal{GP}\left(\mathbf{y}^F(\boldsymbol{\xi}) \,\middle|\, \mathbf{0}, \sum_{f=1}^{F}\left[k^f(\boldsymbol{\xi}, \boldsymbol{\xi}'|\boldsymbol{\theta}^f) \otimes \boldsymbol{\Omega}^f + \delta(\boldsymbol{\xi}, \boldsymbol{\xi}')\tau^f\mathbf{I}\right]\right), \tag{6}$$

by virtue of the independence assumption. Again, this is the GP over the noisy observations with cumulative i.i.d. noise variance $\sum_f \tau_f$, and underlying latent function $\mathbf{y}_l^F(\boldsymbol{\xi}) \sim \mathcal{GP}(\mathbf{y}_l^F(\boldsymbol{\xi}) \mid \mathbf{0}, \sum_{f=1}^{F} k^f(\boldsymbol{\xi}, \boldsymbol{\xi}'|\boldsymbol{\theta}^f) \otimes \boldsymbol{\Omega}^f)$.

Beginning with the lowest fidelity level $f = 1$, we assume the prior (3) and use inputs $\mathcal{X}^1$ together with outputs $\mathbf{Y}^1$ to learn $\{\tau^1, \boldsymbol{\theta}^1\}$ and find the predictive posterior $\mathbf{r}_l^1(\boldsymbol{\xi}) \sim \mathcal{N}(\mathbf{r}_l^1(\boldsymbol{\xi}) \mid \boldsymbol{\mu}_r^1(\boldsymbol{\xi}), \mathbf{V}_r^1(\boldsymbol{\xi}))$. At the next step we use inputs $\mathcal{X}^2$ and outputs $\mathbf{r}^2 := \mathbf{Y}^2 - \mathbf{Y}_{\mathbf{e}_2}^1$ in the same procedure to learn $\{\tau^2, \boldsymbol{\theta}^2\}$, and obtain the posterior for $\mathbf{r}_l^2(\boldsymbol{\xi}) \sim \mathcal{N}(\mathbf{r}_l^2(\boldsymbol{\xi}) \mid \boldsymbol{\mu}_r^2(\boldsymbol{\xi}), \mathbf{V}_r^2(\boldsymbol{\xi}))$. Here we define a matrix of residuals $\mathbf{R}^2$ using the notation introduced in Section 2. This procedure is repeated up to fidelity level $f = F$, i.e., independently learning the hyperparameters associated with each $\mathbf{r}^f(\boldsymbol{\xi})$, given observations $\{\mathbf{R}^f = \mathbf{Y}^f - \mathbf{Y}_{\mathbf{e}_f}^{f-1}\}_{f=2}^{F}$. The predictive posteriors over $\mathbf{r}_l^f(\boldsymbol{\xi})$ are derived using standard Gaussian conditioning rules [7], and the posterior for the high-fidelity latent function $\mathbf{y}_l^F(\boldsymbol{\xi})$ can be written compactly as the following sum of GPs

$$\begin{aligned} \mathbf{y}_l^F(\boldsymbol{\xi}) &= \sum_{f=1}^{F} \mathbf{r}_l^f(\boldsymbol{\xi}) \sim \mathcal{N}(\mathbf{y}_l^F(\boldsymbol{\xi}) \mid \boldsymbol{\mu}^F(\boldsymbol{\xi}), \mathbf{V}^F(\boldsymbol{\xi})), \\ \boldsymbol{\mu}^F(\boldsymbol{\xi}) &= \sum_{f=1}^{F} \boldsymbol{\mu}_r^f(\boldsymbol{\xi}) = \sum_{f=1}^{F} (\boldsymbol{\Omega}^f \otimes \mathbf{k}^f(\boldsymbol{\xi}))^T (\boldsymbol{\Omega}^f \otimes \mathbf{k}^f + \tau^f\mathbf{I})^{-1}\mathrm{vec}(\mathbf{R}^f), \\ \mathbf{V}^F(\boldsymbol{\xi}) &= \sum_{f=1}^{F} \mathbf{V}_r^f(\boldsymbol{\xi}) = \sum_{f=1}^{F} [\boldsymbol{\Omega}^f \otimes k^f(\boldsymbol{\xi}, \boldsymbol{\xi}|\boldsymbol{\theta}^f) - (\boldsymbol{\Omega}^f \otimes \mathbf{k}^f(\boldsymbol{\xi}))^T (\boldsymbol{\Omega}^f \otimes \mathbf{k}^f + \tau^f\mathbf{I})^{-1}(\boldsymbol{\Omega}^f \otimes \mathbf{k}^f(\boldsymbol{\xi}))], \end{aligned} \tag{7}$$

where $\mathrm{vec}(\cdot)$ denotes vectorization and $\mathbf{k}^f(\boldsymbol{\xi}) = (k^f(\boldsymbol{\xi}, \boldsymbol{\xi}_1|\boldsymbol{\theta}^f), \ldots, k^f(\boldsymbol{\xi}, \boldsymbol{\xi}_{N_f}|\boldsymbol{\theta}^f))^T$ is the vector of covariances between the latent function values $\mathbf{r}_l^f(\cdot)$ at $\boldsymbol{\xi}$ and points in $\mathcal{X}^f$. We note that since the posteriors over each $\mathbf{r}^f(\boldsymbol{\xi})$ are learned independently, the training procedure is parallelizable if implemented without active learning (discussed below).

The formulation (7) considers a general case in which noise terms $\tau^f \mathbf{I}$ are included. A computational model (of any fidelity) can, however, be treated as deterministic function without random noise [23]. Thus, we can consider the residual information to be deterministic and model it using GPs without the noise terms, in which case $\mathbf{y}_l^F(\boldsymbol{\xi}) = \mathbf{y}^F(\boldsymbol{\xi})$ and the predictive posterior reduces to

$$
\begin{aligned}
\mathbf{y}^F(\boldsymbol{\xi}) &\sim \mathcal{N}\big(\mathbf{y}^F(\boldsymbol{\xi}) \mid \boldsymbol{\mu}^F(\boldsymbol{\xi}), \mathbf{V}^F(\boldsymbol{\xi})\big), \\
\boldsymbol{\mu}^F(\boldsymbol{\xi}) &= \sum_{f=1}^{f} \boldsymbol{\mu}_r^f(\boldsymbol{\xi}) = \sum_{f=1}^{F} (\boldsymbol{\Omega}^f \otimes \mathbf{k}^f(\boldsymbol{\xi}))^T (\boldsymbol{\Omega}^f \otimes \mathbf{k}^f)^{-1} \mathrm{vec}(\mathbf{R}^f) \\
&= \sum_{f=1}^{F} \mathbf{I} \otimes (\mathbf{k}^f(\boldsymbol{\xi}))^T (\mathbf{k}^f)^{-1} \mathrm{vec}(\mathbf{R}^f) \\
&= \sum_{f=1}^{F} (\mathbf{k}^f(\boldsymbol{\xi}))^T (\mathbf{k}^f)^{-1} \mathbf{R}^f, \\
\mathbf{V}^F(\boldsymbol{\xi}) &= \sum_{f=1}^{f} \mathbf{V}_r^f(\boldsymbol{\xi}) = \sum_{f=1}^{F} \boldsymbol{\Omega}^f \otimes \big(k^f(\boldsymbol{\xi}, \boldsymbol{\xi} | \boldsymbol{\theta}^f) - (\mathbf{k}^f(\boldsymbol{\xi}))^T (\mathbf{k}^f)^{-1} \mathbf{k}^f(\boldsymbol{\xi})\big).
\end{aligned}
\tag{8}
$$

Note that the coregionalization matrices $\boldsymbol{\Omega}^f$ cancel out for the expectation predictions (of any fidelity). This is consistent with autokrigeability [24], which is utilized by Xing et al. [23] to deal with high-dimensional, single-fidelity mechanical design simulations. Noiseless data is the usual assumption for simulations (the so-called ground truth approximation), but one could attempt to incorporate systematic errors arising from the model formulation (model inadequacy), from the numerical approximation (numerical errors), or from parameter uncertainty.

In the same spirit as Xing et al. [23], we now simplify the model by setting $\boldsymbol{\Omega}^f = \mathbf{I}$ for $f = 1, \ldots, F$. This assumes that the components of $\mathbf{y}^F(\boldsymbol{\xi})$ are mutually independent given $\{\boldsymbol{\theta}^f\}$. Note that the variance of the prediction then ignores the spatial correlations, which will affect the uncertainty estimation, but the mean prediction is unaffected. We will discuss this issue further in Section 3.2, in which we see that the active learning process is not affected.

Retaining the noise terms and coregionalisation matrices, i.e., retaining Eq. (7), requires an additional set of hyperparameters, namely the τ^f and the entries of each $\boldsymbol{\Omega}^f$. We may even use a richer covariance structure, such as the linear model of coregionalisation [25] with linear combinations of separable covariances. In low-dimensional output spaces, this could be of benefit in terms of accuracy and variance capture. In very high-dimensional spaces, on the other hand, these approaches lead to problems in terms of maximizing the likelihood (or evidence lower bound) due to the higher number of inputs in the optimization problem [25] (see Section 3.3). If modelling the noise is deemed important for a particular problem (perhaps because the solutions are known to be corrupted), a compromise could be achieved by retaining the $\{\tau^f\}$ terms and setting $\boldsymbol{\Omega}^f = \mathbf{I}$. We discuss this formulation later. We point out, however, that in the examples in Section 4, the noise terms are not included.

For most data sets, N_f decreases in size with f, and is typically small for large values of f. The uncertainty in the predictions for $\mathbf{r}_l^f$ (or $\mathbf{r}^f$) therefore increases with f. We have to keep in mind, however, that that uncertainty is also bounded by the scales of the residuals between different fidelities. For high values of f, the simulations for adjacent fidelities would tend to be similar, and therefore the residuals would tend to be small. Thus, the added uncertainties for large f will be small. For small f, the residuals between adjacent fidelities are expected to be larger but the uncertainties would be smaller by virtue of the greater number of samples. The additive structure can, therefore, potentially ensure that the final uncertainty reflects the true model uncertainty. We discuss this further in the first example in section 4.1.

3.1. Model training and high-fidelity predictions

Given the residual information $\mathbf{R}^f$, we can derive the residual marginal log-likelihood at fidelity f

$$
\begin{aligned}
\mathcal{L}^f &= -\frac{1}{2} |\mathbf{I} \otimes \mathbf{k}^f| - \frac{1}{2} \mathrm{vec}(\mathbf{R}^f)^T (\mathbf{I} \otimes \mathbf{k}^f)^{-1} \mathrm{vec}(\mathbf{R}^{(f)}) - \frac{N_f}{2} \ln(2\pi), \\
&= -\frac{d}{2} |\mathbf{k}^f| - \frac{1}{2} \mathrm{tr}((\mathbf{R}^f)^T (\mathbf{k}^f)^{-1} \mathbf{R}^f) - \frac{N_f}{2} \ln(2\pi),
\end{aligned}
\tag{9}
$$

while the joint marginal log-likelihood is

$$
\mathcal{L} = \sum_{f=1}^{F} \mathcal{L}^f = \sum_{f=1}^{F} -\frac{d}{2} |\mathbf{k}^f| - \frac{1}{2} \mathrm{tr}((\mathbf{R}^f)^T (\mathbf{k}^f)^{-1} \mathbf{R}^f) - \frac{N_f}{2} \ln(2\pi).
\tag{10}
$$

The availability of residual information $\mathbf{R}^f$ for each fidelity relies on the fact that $\mathbf{X}^f \subset \mathbf{X}^{f-1}$. The marginal log likelihoods $\mathcal{L}^f$ are independent from each other given the inputs, and thus the training process can be performed separately for each fidelity, i.e., parallelised. Making predictions using ResGPis straightforward because the posterior for fidelity F is the sum of

the posteriors over the residuals, each of which is Gaussian. The prediction is given by Eq. (8) with

$$\mathbf{V}^F(\boldsymbol{\xi}) = \sum_{f=1}^{f} \mathbf{V}_r^f(\boldsymbol{\xi}) = \sum_{f=1}^{F} \mathbf{I} \otimes \left(k^f(\boldsymbol{\xi}, \boldsymbol{\xi}|\boldsymbol{\theta}^f) - (\mathbf{k}^f(\boldsymbol{\xi}))^T (\mathbf{k}^f)^{-1} \mathbf{k}^f(\boldsymbol{\xi}) \right). \tag{11}$$

The condition $\mathcal{X}^F \subset \mathcal{X}^{F-1} \subset \ldots \subset \mathcal{X}^1$ should in practice not be an issue since the experimental design can be chosen *a priori*. If for any reason this condition is not satisfied, the fidelity $f-1$ posterior GP

$$\mathbf{y}_l^{f-1}(\boldsymbol{\xi}) = \sum_{k=1}^{f-1} \mathbf{r}^k \tag{12}$$

can be used to approximate $\mathbf{y}_l^{f-1}(\boldsymbol{\xi})$ at the inputs in $\mathcal{X}^f$, which can then used to define the residual data $\mathbf{R}^f$ within the ResGP framework. In such a case, it may be beneficial to include the noise terms to account for the approximation error in the residual (possibly setting $\tau^1 = 0$). If we assume $\boldsymbol{\Omega}^f = \mathbf{I}$, this leads to

$$\begin{aligned}
\mathbf{y}_l^F(\boldsymbol{\xi}) &= \sum_{f=1}^{F} \mathbf{r}_l^f(\boldsymbol{\xi}) \sim \mathcal{N}\left(\mathbf{y}_l^F(\boldsymbol{\xi}) \,|\, \boldsymbol{\mu}^F(\boldsymbol{\xi}), \mathbf{V}^F(\boldsymbol{\xi}) \right), \\
\boldsymbol{\mu}^F(\boldsymbol{\xi}) &= \sum_{f=1}^{F} (\mathbf{k}^f(\boldsymbol{\xi}))^T (\mathbf{k}^f + \tau^f \mathbf{I})^{-1} \mathbf{R}^f, \\
\mathbf{V}^F(\boldsymbol{\xi}) &= \sum_{f=1}^{F} \mathbf{I} \otimes \left(k^f(\boldsymbol{\xi}, \boldsymbol{\xi}|\boldsymbol{\theta}^f) - (\mathbf{k}^f(\boldsymbol{\xi}))^T (\mathbf{k}^f + \tau^f \mathbf{I})^{-1} \mathbf{k}^f(\boldsymbol{\xi}) \right).
\end{aligned} \tag{13}$$

Such a model is still highly scalable with the output space dimension d, requiring only F additional parameters compared to the basic ResGP(see Section 3.3). We do not consider this model in the examples. The efficacy of such an approach depends upon the disparity between the sets $\mathcal{X}^f$, and requires a deeper investigation that is beyond the scope of the present work.

3.2. Active learning via variance reduction

In practice, data of high fidelity is expensive to obtain. It is desirable to allocate computational resources, especially for the high-fidelity simulations, such that the surrogate model can achieve its best performance with the least computational cost. We first define the *information gain* for fidelity f at a new parameter $\boldsymbol{\xi}_*$ as the uncertainty or variance given the current data collection $\overline{\mathcal{X}}$. Maximization of the information gain can then be defined by

$$\boldsymbol{\xi}_*^f = \underset{\boldsymbol{\xi} \in \overline{\mathcal{X}}}{\operatorname{argmax}} \ \operatorname{tr}\left(\mathbf{V}_r^f(\boldsymbol{\xi}) \right), \tag{14}$$

in which tr denotes the trace operator. The coregionalization matrices in Eq. (8) are irrelevant as far as the information gain is concerned for any given $\boldsymbol{\xi}$ since

$$\begin{aligned}
\operatorname{tr}\left(\mathbf{V}_r^f(\boldsymbol{\xi}) \right) &= \operatorname{tr}\left(\boldsymbol{\Omega}^f \otimes \left(k^f(\boldsymbol{\xi}, \boldsymbol{\xi}|\boldsymbol{\theta}^f) - (\mathbf{k}^f(\boldsymbol{\xi}))^T (\mathbf{k}^f)^{-1} \mathbf{k}^f(\boldsymbol{\xi}) \right) \right) \\
&= \operatorname{tr}\left(\boldsymbol{\Omega}^f \right)\left(k^f(\boldsymbol{\xi}, \boldsymbol{\xi}|\boldsymbol{\theta}^f) - (\mathbf{k}^f(\boldsymbol{\xi}))^T (\mathbf{k}^f)^{-1} \mathbf{k}^f(\boldsymbol{\xi}) \right),
\end{aligned} \tag{15}$$

and $\operatorname{tr}(\boldsymbol{\Omega}^f)$ is treated as a constant in the maximization. Thus, setting $\boldsymbol{\Omega}^f = \mathbf{I}$ does not affect our model either in terms of making predictions or in terms of utilizing the uncertainty for active learning or Bayesian optimization. For decision making or active learning, the uncertainty related to a new sample (for a new input) rather than to a particular component of a new sample is essential. Eq. (14) now reduces to

$$\boldsymbol{\xi}_*^f = \underset{\boldsymbol{\xi} \in \overline{\mathcal{X}}}{\operatorname{argmax}} \ \left(k^f(\boldsymbol{\xi}, \boldsymbol{\xi}|\boldsymbol{\theta}^f) - (\mathbf{k}^f(\boldsymbol{\xi}))^T (\mathbf{k}^f)^{-1} \mathbf{k}^f(\boldsymbol{\xi}) \right). \tag{16}$$

Inspired by the work of Narayan et al. [17], we propose to build the multi-fidelity surrogate model in a sequential manner, starting from the lowest fidelity. For each fidelity, based on the available data, we train the GP model and compute the information gain for each candidate input. Subsequently, the fidelity f experiments corresponding to the candidate inputs that yield the maximum information gain are conducted and added to the training data. This process is repeated until a given condition is met. We present the full details of how to construct the model without requiring the prior execution of low-fidelity simulations for all candidates (which is required by the classic stochastic collocation model) in Algorithm 1. In this algorithm, the stopping criteria is a given number of simulation runs for each fidelity, which should be decided based on the available computational budget. In the case that we require the system to be fully automatic, we may instead specify a large candidate set $\mathcal{X}_t$ and the uncertainty bound for determining the number of iterations. We can also perform an eigenvalue analysis of each correlation matrix to find the optimal number of samples that fully capture the model behaviour within the parameter space. We note that there are more state-of-the-art active learning methods, such as the work of Song et al. [26], which utilizes mutual information across different fidelities. The active learning component is not the main focus of this work, but such recent developments could improve the performance of ResGP. We also point out that parallelization of the training process is not straightforward if active learning is implemented.

Algorithm 1 ResGPSequential Construction

Require: A finite-cardinality set $\mathcal{X}_t$ of inputs, and thenumber of experiments allowed for each fidelity N_f
Ensure: A trained ResGP model

    Randomly select a point from $\mathcal{X}_t$ and form the initial 1-fidelity candidate set $\mathcal{X}^1$
    Conduct 1-fidelity experiment for $\mathcal{X}^1$ and collect the solution in matrix $\mathbf{Y}^1$
    Update residual matrix $\mathbf{R}^1 = \mathbf{Y}^1$
    for $|\mathcal{X}^1| < N_1$ **do**
        Update 1-fidelity residual GP model by MLE (9) ($f = 1$) based on inputs $\mathcal{X}^1$ and residuals $\mathbf{R}^1$
        Find $\boldsymbol{\xi}_*$ based on Eq. (16) for $\mathcal{X}_t \setminus \mathcal{X}^1$
        Update 1-fidelity candidate set $\mathcal{X}^1 \leftarrow \mathcal{X}^1 \cup \{\boldsymbol{\xi}_*\}$
        Conduct a 1-fidelity simulation to obtain solution $\mathbf{y}^1(\boldsymbol{\xi}_*)$
        Update low-fidelity solution matrix $\mathbf{R}^1 \leftarrow [\mathbf{R}^1, \mathbf{y}^1(\boldsymbol{\xi}_*)]$
    end for
    for $f = 2, \ldots, F$ **do**
        Randomly select a point from $\mathcal{X}^{f-1}$ to form initial candidate set $\mathcal{X}^f$
        Conduct f-fidelity experiment for $\mathcal{X}^f$ and collect the solution in matrix $\mathbf{Y}^f$
        Update residual matrix $\mathbf{R}^f = \mathbf{Y}^f - \mathbf{Y}^{f-1}_{\mathbf{e}_f}$
        for $|\mathcal{X}^f| < N_f$ **do**
            Update f-fidelity residual GP model by MLE of (9) based on $\mathcal{X}^f$, $\mathbf{R}^f$
            Find $\boldsymbol{\xi}_*$ based on Eq. (16) for $\mathcal{X}^{f-1} \setminus \mathcal{X}^f$
            Update f-fidelity candidate set $\mathcal{X}^f \leftarrow \mathcal{X}^f \cup \{\boldsymbol{\xi}_*\}$
            Conduct an f-fidelity simulation to obtain solution $\mathbf{y}^f(\boldsymbol{\xi}_*)$
            Update f-fidelity residual matrix $\mathbf{R}^f \leftarrow [\mathbf{R}^f, \mathbf{y}^f(\boldsymbol{\xi}_*) - \mathbf{y}^{f-1}(\boldsymbol{\xi}_*)]$
        end for
    end for

Table 1
Model complexity comparison.

Method	Complexity	Number of parameters
ResGPwith $\boldsymbol{\Omega}^f = \mathbf{I}, \sigma^f = 0$	$\mathcal{O}(\sum_{f=1}^{F} N_f^3)$	$F(l+1)$
ResGPwith $\boldsymbol{\Omega}^f = \mathbf{I}, \sigma^f \neq 0$	$\mathcal{O}(\sum_{f=1}^{F} N_f^3)$	$F(l+2)$
ResGPwith $\boldsymbol{\Omega}^f \neq \mathbf{I}, \sigma^f \neq 0$	$\mathcal{O}(\sum_{f=1}^{F} N_f^3 d^3)$	$F(l+2+d(d+1)/2)$
NARGP	$\mathcal{O}(\sum_{f=1}^{F} N_f^3)$	$F(l+1) + (F-1)d$
MF-DGP	$\mathcal{O}(FN_T M^2 d)$	$(F-1)(Md(2 + \frac{1}{2}(Md+1)) + l(M+1) + 1 + d) + Md + l + 1$
AR1	$\mathcal{O}(N_T^3)$	$F(l+2) - 1$
Greedy NARGP	$\mathcal{O}(\sum_{f=1}^{F} N_f^3)$	$(l+1) + (F-1)d$

3.3. Computational complexity

For standard GP models with N training samples, the time (computational) complexity for model training is $\mathcal{O}(N^3)$ due to the inversion of an $N \times N$ covariance matrix in the maximum log-likelihood solution (see Eq. (9)) [21]. Given its structure, as the sum of conditionally independent GPs, ResGPcan scale well with the output dimension. The computational complexity is that for the F GPs with $N_f, f = 1, \ldots, F$, training samples, namely $\mathcal{O}(\sum_{f=1}^{F} N_f^3)$. The total number of model parameters for ResGPis $F(l+1)$ with a standard ARD kernel. This follows from the form of the ARD kernel (5), which requires $l+1$ hyperparameters for each f. With added noise, an additional hyperparameter σ^f is required for each GP, leading to $F(l+2)$ hyperparameters. If $\boldsymbol{\Omega}^f \neq \mathbf{I}$, the covariance matrix is of size $N_f d \times N_f d$ for each f, and the entries of each symmetric $d \times d$ matrix $\boldsymbol{\Omega}^f$ need to be estimated, leading to an additional $Fd(d+1)/2$ parameters.

The computational complexity comparison with other state-of-the-art methods is shown in Table 1 (for ARD kernels), in which M is the number of inducing points for MF-DGP and $N_T = \sum_{f=1}^{F} N_f$ is the total number of training samples. For AR, the covariance matrix is of size $N_T \times N_T$ [2] [section 2.3], thus leading to the computational complexity shown in Table 1. The number of hyperparameters is $F(l+1)$ for F ARD kernels, with an additional $F-1$ hyperparameters for constants defining the relationships between successive fidelities. NARGP involves F GPs with covariance matrices of sizes $N_f \times N_f$, leading to a computational complexity equal to that of ResGP[14]. Since the outputs $\mathbf{y}^{f-1}(\boldsymbol{\xi})$ are treated as inputs alongside $\boldsymbol{\xi}$ for fidelity $f = 2, \ldots, F$, there are an additional $(F-1)d$ hyperparameters compared to ResGPwith $\boldsymbol{\Omega}^f = \mathbf{I}, \sigma^f = 0$.

Greedy NAR also involves F GP training steps with $N_f \times N_f$ covariance matrices [18]. Like NAR, it uses the $f-1$ fidelity outputs as inputs for fidelity $f = 2, \ldots, F$, but only involves the model inputs $\boldsymbol{\xi}$ in the lowest fidelity GP, leading to $(F-1)d + l + 1$ hyperparameters for ARD kernels. Since MF-DGP is equivalent to a deep GP with F layers and outputs in $d-$dimensional space in each layer, the computational complexity is dominated by the KullbackLeibler divergence in the

evidence lower bound (ELBO) for the variational approximation, namely $\mathcal{O}(FN_TM^2d)$ [27]. The sparse variational approximation in MF-DGP requires $Md(F-1)+(F-1)Md(Md+1)/2$ variational parameters for the means and symmetric covariance matrices of the distributions over the inducing points (in $\mathbb{R}^d$) for fidelities $f=2,\ldots,F$, and a further $Md+(F-1)M(d+l)$ variational parameters for the inducing inputs at fidelities $f=1,\ldots,F$ [20]. The kernel hyperparameters for an ARD kernel are $l+1$ for fidelity 1 and $(F-1)(l+1+d)$ for the other fidelities, and a further F hyperparameters are included for the noise variances. SC does not contain any model parameters and thus no model training is required. It only involves a single computation of the inverse of the Gram matrix at low fidelity [17]. It does, however, require low-fidelity experiments in order to make predictions at high fidelity, which is usually far more costly.

The computational complexity for MF-DGP is prohibitive for high-dimensional problems, and AR also suffers from high costs along with ResGPwith $\boldsymbol{\Omega}^f\neq\mathbf{I}$. The numbers of parameters for NARGP, Greedy NAR and MF-DGP are also excessive, which can mean that training is problematic for large d. The same is true for ResGPwithout the assumption $\boldsymbol{\Omega}^f=\mathbf{I}$; setting $\boldsymbol{\Omega}^f=\mathbf{I}$ and retaining the noise terms, however, leads again to a highly scalable model.

W-Xing: I have sent earlier

3.4. Error bounds for the univariate ResGP

In this section we will prove an error bound on ResGP in the univariate case. We begin with some definitions. A symmetric kernel function $k:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ is positive semi definite (psd) if the corresponding matrix $[k(\boldsymbol{\xi}_n,\boldsymbol{\xi}_m)]_{nm}$ for any finite set $\{\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_N\}\subset\mathcal{X}\subset\mathbb{R}^l$ is psd. Henceforth, we consider only kernels k that are symmetric, psd and bounded on $\mathcal{X}$. A real-valued function $f:\mathcal{X}\to\mathbb{R}$ is Lipschitz continuous with Lipschitz constant $L\geq0$ if $|f(\boldsymbol{\xi})-f(\boldsymbol{\xi}')|\leq L\|\boldsymbol{\xi}-\boldsymbol{\xi}'\|$, $\forall\boldsymbol{\xi},\boldsymbol{\xi}'\in\mathcal{X}\subset\mathbb{R}^l$, in which the standard Euclidean norm is used ($\|\cdot\|$ is used to denote a standard Euclidean norm throughout). We define a kernel k to be Lipschitz continuous with a Lipschitz constant L^k in the sense that

$$|k(\boldsymbol{\xi},\overline{\boldsymbol{\xi}})-k(\boldsymbol{\xi}',\overline{\boldsymbol{\xi}})|\leq A(\overline{\boldsymbol{\xi}})\|\boldsymbol{\xi}-\boldsymbol{\xi}'\|\leq\sup_{\overline{\boldsymbol{\xi}}}A(\overline{\boldsymbol{\xi}})\|\boldsymbol{\xi}-\boldsymbol{\xi}'\|:=L^k\|\boldsymbol{\xi}-\boldsymbol{\xi}'\|,\quad\forall\boldsymbol{\xi},\boldsymbol{\xi}',\overline{\boldsymbol{\xi}}\in\mathcal{X}. \tag{17}$$

Most commonly used kernels, including the squared-exponential and the Matern class of kernels, are Lipschitz continuous in the sense defined above. A function $f:\mathcal{X}\to\mathbb{R}$ admits a monotonic function $\omega(\cdot)$ as a modulus of continuity iff $|f(\boldsymbol{\xi})-f(\boldsymbol{\xi}')|\leq\omega(\|\boldsymbol{\xi}-\boldsymbol{\xi}'\|)$, $\forall\boldsymbol{\xi},\boldsymbol{\xi}'\in\mathcal{X}$. The τ covering number $M(\tau,\mathcal{X})$ of a set $\mathcal{X}$ is defined as the minimum number of open balls with radius τ (with respect to the standard Euclidean metric) that is required to completely cover $\mathcal{X}$. We use $\|\mathbf{A}\|_2=\sup_{\|\mathbf{x}\|=1}\|\mathbf{Ax}\|$ to denote the matrix norm of $\mathbf{A}\in\mathbb{R}^{n\times m}$ induced by the standard Euclidean norm in $\mathbb{R}^m$.

A number of results have been obtained in relation to error bounds for the simplest univariate GP models. Most of the bounds are derived based on the theory of reproducing kernel Hilbert spaces (RKHS) (a recent review can be found in [28]). For every psd kernel $k(\cdot,\cdot)$ there exists a unique RKHS $\mathcal{H}_k$, the functions f in which inherit the smoothness properties of the kernel. This is easily seen by the following characterization of a RKHS: define an inner product space $\mathcal{V}$

$$\mathcal{V}=\left\{f:f(\cdot)=\sum_{n=1}^{\infty}a_nk(\cdot,\boldsymbol{\xi}_n),\quad a_n\in\mathbb{R},\ \boldsymbol{\xi}_n\in\mathcal{X}\right\} \tag{18}$$

with inner product $\langle f,g\rangle_{\mathcal{H}_k}=\sum_n\sum_m a_nb_mk(\boldsymbol{\xi}_n,\boldsymbol{\xi}'_m)$, where $g(\cdot)=\sum_m b_mk(\cdot,\boldsymbol{\xi}'_m)$, and induced norm $\|f\|_{\mathcal{H}_k}=\langle f,f\rangle_{\mathcal{H}_k}^{1/2}$. $\mathcal{V}$ is a pre-Hilbert space from which we obtain the unique RKHS as $\mathcal{H}_k=\{f\in\overline{\mathcal{V}}:\|f\|_{\mathcal{H}_k}<\infty\}$, where the closure is defined with respect to the metric induced by $\|\cdot\|_{\mathcal{H}_k}$. For the squared exponential and other common kernels, $\mathcal{H}_k\subset C^\infty(\mathcal{X})$, while for Matern kernels, $\|\cdot\|_{\mathcal{H}_k}$ is norm-equivalent to a Sobolev space $W^{s,2}(\mathcal{X})$, $s>l/2$. In general, therefore, RKHSs are quite restrictive, and are small compared to the support of a prior GP distribution with the covariance function (kernel) that defines the RKHS.

Bounds can be derived for the GP regression error using the equivalence between GP regression and stationary kernel interpolation [29], the latter of which can be posed as an optimization problem in the RKHS $\mathcal{H}_k$ corresponding to the interpolation kernel k [28]. If the true function in GP modelling is hypothesised to lie in this space, the results can be carried over. The bounds are given in terms of a power function, which is identified with the posterior variance of an equivalent GP model. For noisy data, analogous results for kernel ridge regression can be used, in which case the error bounds depend on the norm $\|f\|_{\mathcal{H}_k}$ of the unknown function f, as well as an empirical covering number with respect to the $L^2(\mu)$ norm, where μ is the unknown distribution over the data [30]. Again for noisy observations, information theoretic and RKHS approaches were used by Srinivas et al. [31], later improved upon by Chowdhury and Gopalan [32], to find error bounds for GP regression. These bounds involve constants that in practice are difficult to obtain. An alternative hypothesis is to take the support of the prior distribution of the GP as the belief space from which to seek the true function. This hypothesis has been employed in stochastic bandit problems based on GPs [33,34] and more recently has been used to establish general interpretable bounds for basic GP models [19,35]. The sample space is the largest possible space of candidate functions, and leads to bounds that can be approximated for common settings with relative ease, in comparison to the RKHS approaches. In the analysis below, we use the interpolation (noise-free) results of Wang et al. [34], and, in particular, follow closely the analysis of Lederer et al. [19] to extend their univariate, single-fidelity probabilistic uniform error bound to ResGP. Such a bound is defined as follows

Definition 3.1. A GP estimate $\mu(\boldsymbol{\xi})$ of an unknown function $y(\boldsymbol{\xi})$ has a uniformly bounded error on a compact set $\mathcal{X} \subset \mathbb{R}^l$ if there exists a function $g(\boldsymbol{\xi})$ such that

$$|\mu(\boldsymbol{\xi}) - y(\boldsymbol{\xi})| \leq g(\boldsymbol{\xi}), \quad \forall \boldsymbol{\xi} \in \mathcal{X}. \tag{19}$$

If this bound holds with probability of at least $1 - \delta$ for some $\delta \in (0, 1)$, it is called a probabilistic uniform error bound.

As stated above, we require the following main assumption over the unknown function, in this case $y^F(\boldsymbol{\xi})$, namely that it belongs to the sample space of the prior GP.

Assumption 3.2. $y^F(\boldsymbol{\xi})$ is a realisation (is in the sample space) of the following GP

$$\mathcal{GP}\left(y^F(\boldsymbol{\xi}) \mid 0, \sum_{f=1}^{F} k^f(\boldsymbol{\xi}, \boldsymbol{\xi}'|\boldsymbol{\theta}^f)\right), \tag{20}$$

and observations y_n^F are evaluations of this function at the design points, that is $y_n^F = y^F(\boldsymbol{\xi}_n)$, $\forall \boldsymbol{\xi}_n \in \mathcal{X}^F$.

The properties of sample paths of this GP (the belief space) are again closely related to the smoothness of the kernel, but are more difficult to establish and quantify. For stationary kernels $k(\boldsymbol{\xi} - \boldsymbol{\xi}')$, almost surely (a.s.) or sample path continuity is ensured by sufficient smoothness of k at the origin [36] (section 2.5). Similar continuity results can be established for derivatives of the sample paths, essentially requiring $2k$ times differentiability of the kernel to establish k times differentiability of sample paths [36] (section 2.5.2). Although Lipschitz continuity is a rather strong form a continuity, implying amongst other things uniform continuity, this assumption is still rather mild in comparison to the RKHS hypothesis. The following Lemma (see Appendix A for the proof) enables us to prove the main result on the uniform error bound for the ResGP approximation of y^F.

Lemma 3.3. _Consider the posterior GP process defined by (20), in which all of the kernels $k^f(\boldsymbol{\xi}, \boldsymbol{\xi}'|\boldsymbol{\theta}^f)$ are assumed to be Lipschitz continuous with Lipschitz constants L_k^f on the compact sets $\mathcal{X}^f \subset \mathcal{X}$, $f = 1, \ldots, F$. Furthermore, consider a continuous unknown function $y^F : \mathcal{X} \to \mathbb{R}$ with Lipschitz constant L_y and N_F observations y_n^F satisfying Assumption 3.2. Then the posterior mean $\mu^F(\cdot)$ defined in (8) and the standard deviation $\sigma^F(\cdot) = \sqrt{v^F(\boldsymbol{\xi})}$ (Eq. (11)) of the GP conditioned on $\{(\boldsymbol{\xi}_n, y_n^F)\}_{n=1}^{N_F}$ are both continuous, with Lipschitz constant L_{μ^F} and modulus of continuity $\omega_\sigma^F(\cdot)$ on $\mathcal{X}$, respectively, satisfying_

$$L_{\mu^F} \leq \sum_{f=1}^{F} L_k^f \sqrt{N_f} \|(\mathbf{k}^f)^{-1}\mathbf{R}^f\| \tag{21}$$

$$\omega_\sigma^F(\|\boldsymbol{\xi} - \boldsymbol{\xi}'\|) \leq \left[\sum_{f=1}^{F} 2L_k^f\left(1 + N_f \max_{\boldsymbol{\xi}, \boldsymbol{\xi}' \in \mathcal{X}} k^f(\boldsymbol{\xi}, \boldsymbol{\xi}'|\boldsymbol{\theta}^f) \|(\mathbf{k}^f)^{-1}\|_2\right)\|\boldsymbol{\xi} - \boldsymbol{\xi}'\|\right]^{1/2}. \tag{22}$$

Remark 3.4. Any kernel that is everywhere differentiable and has bounded partial derivatives is Lipschitz continuous and any Lipschitz constant satisfies $L_k^f \leq \sup_{\boldsymbol{\xi}, \boldsymbol{\xi}' \in \mathcal{X}} \|\nabla_{\boldsymbol{\xi}} k(\boldsymbol{\xi}, \boldsymbol{\xi}')\|_\infty$, where $\| \cdot \|_\infty$ is the l^∞ norm. The Lipschitz constant L_y is clearly not available in practice. However, a probabilistic bound can be obtained as in Theorem 3.2 in [19]. The remaining terms in these expressions depend only on the training data and kernels, which are explicitly known. Using Cauchy-Schwartz and the fact that $\| \cdot \|_2$ is subordinate to $\| \cdot \|$, the bound for L_{μ^F} can be elaborated further as follows

$$\sum_{f=1}^{F} L_k^f \sqrt{N_f} \|(\mathbf{k}^f)^{-1}\mathbf{R}^f\| \leq \sum_{f=1}^{F} L_k^f \sqrt{N_f} \frac{\|\mathbf{R}^f\|}{\rho_m(\mathbf{k}^f)}, \tag{23}$$

in which $\rho_m(\mathbf{k}^f) = \|(\mathbf{k}^f)^{-1}\|_2^{-1}$ is the smallest singular value of $\mathbf{k}^f$. $\rho_m(\mathbf{k}^f) > 0$ $\forall f$ since $\mathbf{k}^f$ is psd.

The error on any design $\mathcal{X}_\tau^F$ is bounded as in Srinivas et al. [31] [Lemma 5.1], introducing a constant $\beta(\tau)$, which depends on an upper bound τ for the fill distance of $\mathcal{X}_\tau^F$, defined as $h = \sup_{\boldsymbol{\xi} \in \mathcal{X}} \min_{\boldsymbol{\xi}' \in \mathcal{X}_\tau^F} \|\boldsymbol{\xi} - \boldsymbol{\xi}'\|$. h represents the radius of the largest ball in $\mathcal{X}$ that does not contain any point in the design $\mathcal{X}_\tau^F$. In greedy designs that minimise the GP posterior variance (as in the active learning component), for any kernel k that induces a RKHS that is norm equivalent to $W^{s,2}(\mathbb{R}^l)$ with $s > l/2$, Santin and Haasdonk [37] showed that $\forall \epsilon > 0$, $\exists C_\epsilon > 0$ such that $h \leq C_\epsilon |\mathcal{X}_\tau^F|^{-1/2+\epsilon}$, in which $|\mathcal{X}_\tau^F|$ is the cardinality of $\mathcal{X}_\tau^F$.

The minimum number of grid points satisfying the upper bound on the fill distance is $M(\tau, \mathcal{X})$, an upper bound for which is $M(\tau, \mathcal{X}) \leq (1 + e/\tau)^d$, assuming a hypercubic set $\mathcal{X}$ with edge length e. By utilizing Lemma 3.3 and the continuity of y^F, the error bound on $\mathcal{X}_\tau^F$ can then be extended to the whole of $\mathcal{X}$, which leads to Theorem 3.5 below

Theorem 3.5 (Lederer et al. [19],Theorem 3.1)**.** _Let the conditions of Lemma 3.3 be satisfied. Pick $\delta \in (0, 1)$, $\tau \in \mathbb{R}$, $\tau > 0$, and set_

$$\beta(\tau) = 2 \log\left(\frac{M(\tau, \mathcal{X})}{\delta}\right) \tag{24}$$

Table 2

Comparisons of the root mean square error (RMSE), R^2 and mean negative log likelihood (MNLL) for ResGP and three state-of-the-art methods on five univariate multi-fidelity benchmark problems. The number of training samples is given in the form $N_1 - N_2$ or $N_1 - N_2 - N_3$. *Values were taken from [20] [Table 1].

BENCHMARK	l	N_f	AR1			NARGP			MF-DGP*			ResGP		
			$R{\wedge}2$	RMSE	MNLL	$R{\wedge}2$	RMSE	MNLL	$R{\wedge}2$	RMSE	MNLL	$R{\wedge}2$	RMSE	MNLL
CURRIN	2	20-5	0.918	0.564	11.136	0.947	0.550	14.364	0.935	0.601	0.763	**0.967**	**0.436**	**0.663**
PARK	4	30-5	0.986	0.552	57.397	0.965	0.883	103.891	0.985	0.565	1.383	**0.990**	**0.397**	**0.290**
BOREHOLE	8	60-10	0.999	0.012	−2.886	0.998	0.019	−2.668	0.999	0.015	−2.031	**1.000**	**0.005**	**−3.996**
BRANIN	2	80-30-10	0.912	0.019	**−4.019**	0.724	0.107	5.487	0.965	0.030	−2.572	**0.998**	**0.008**	−3.675
HARTMANN-3D	3	80-30-10	0.996	0.058	−1.414	0.992	0.083	−0.712	0.994	0.075	−0.731	**0.997**	**0.054**	**−1.496**

$$\gamma(\tau) = \left(L_{\mu^F} + L_y\right)\tau + \sqrt{\beta(\tau)}\,\omega_\sigma^N(\tau). \tag{25}$$

Then, it holds that

$$\mathbb{P}\left(|y^F(\boldsymbol{\xi}) - \mu^F(\boldsymbol{\xi})| \le \sqrt{\beta(\tau)}\sigma^F(\boldsymbol{\xi}) + \gamma(\tau), \forall \boldsymbol{\xi} \in \mathcal{X}\right) \ge 1 - \delta. \tag{26}$$

Remark 3.6. Eq. (26) can be computed for fixed τ and δ given the probabilistic upper bound for L_y. We note that $\beta(\tau)$ grows only logarithmically as τ decreases, which limits the growth of $\gamma(\tau)$ as $\tau \to 0$.

4. Results and discussion

In the results presented below, AR, NAR, MF-DGP, Greedy NAR and SC are implemented as per their original formulations. In all of the GP methods, zero-mean functions are assumed by centering the data, i.e., no regression functions are employed. With the exception of MF-DGP, noiseless data is assumed for all fidelities. The original MF-DGP includes noise at all fidelities. For ResGP, AR and Greedy NAR, ARD kernels are used. For NARGP, the fidelity-1 kernel is an ARD kernel, while the fidelity $f = 2, \ldots, F$ kernels are of the form introduced by Perdikaris et al. in the original implementation [14] [Eq. (2.12)]. Each of these kernels is formed from three ARD kernels by factoring the dependence on $\boldsymbol{\xi}^f$ and $\mathbf{y}^{f-1}(\boldsymbol{\xi})$ and adding a third kernel as a bias term. In MF-DGP, enhanced versions of these kernels (adding an additional linear kernel for the dependence on $\mathbf{y}^{f-1}(\boldsymbol{\xi})$) are employed as in Cutajar et al. [20] [Eq. (11)]. For MF-DGP we used the authors' open source code [38], which is available on Github[1]. For all other methods, we used our own implementations with the settings stated above to generate the results.

4.1. Test problem 1: five univariate benchmark problems

We first examine the univariate case and compare the results to AR, NARGP and MF-DGP. A selection of well-known univariate multi-fidelity problems is used, following [20]. This includes three 2-fidelity and two 3-fidelity examples. The functions considered and the definitions of the fildelities are detailed in Appendix B. A comparison of the accuracy of ResGP (without active learning) with the other methods is shown in Table 2. This table includes values of the root mean square error (RMSE), the coefficient of determination (R^2) and the mean negative log likelihood (MNLL) against 1000 randomly selected test points, following [20]. The MNLL provides the most commonly accepted test for the capture of uncertainty. The number of training points for each fidelity N_f and the number of inputs l (selected randomly) are also given in Table 2.

As can be seen, ResGP outperforms all other methods in all examples in terms of the accuracy. For example, the RMSE is at least 21%, 28%, 58%, 58% and 7% lower than for the other methods on the Currin, Park, Borehole, Branin and Hartmann-3d examples, respectively. The MNLL is at least 13%, 79%, 38% and 6% lower than for the other methods on the Currin, Park, Borehole and Hartmann-3d examples, respectively. In the Branin example, the MNLL is 9% more negative for AR. Given their high model flexibilities, it is possible that MF-DGP and NARGP tend to overfit the data and underestimate the uncertainty in these examples. For ResGP, the uncertainties have an additive structure and increase monotonically. If the scale of the residual is large or the sample number small, the uncertainty will likewise be large. In this sense ResGP is more 'careful' with uncertainty estimations (in some problems it may overestimate the uncertainty).

We note that we were unable to reproduce the numbers stated in [20] using the authors' code [38]. Thus, the values appearing in Table 1 for MF-DGP are taken from the original paper [20] [Table 1]. For all other methods we used the training and test data provided by the authors on Github[1] (which they used to generate the results in [20]).

4.2. Test problem 2: turbulent mixing flow in an elbow-shape pipe

A number of models have been developed to study turbulent flows, ranging from simple one-equation models such as the Sparllat–Almaras model to sophisticated models such as Large Eddy Simulation (LES). The former can be considered

---

[1] https://github.com/EmuKit/emukit/tree/master/emukit/examples/multi_fidelity_dgp
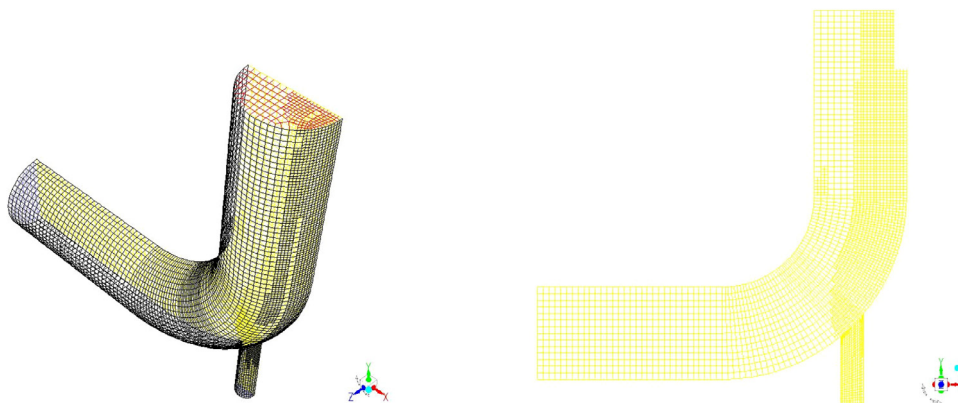
Fig. 1. Computational domain and geometrical configuration of an elbow-shaped pipe and its plane of symmetry (Test Problem 2).

low-fidelity whereas the latter can be considered high-fidelity. For the design and optimization of thermal-fluid systems, high-fidelity models can be impractical. On the other hand, low-fidelity models will lead to sub-optimal designs. Thus, turbulence modelling is prime candidate for combining low- and high-fidelity models.

We applied ResGPwith and without active learning to a benchmark problem of turbulent mixing flow in a pipe and compared it with three state-of-the art methods, namely NARGP, SC (without out-of-sample F1 data, for a fair comparison) and Greedy NAR. As illustrated in Fig. 1, water enters from two inlets, the bottom left end of the pipe and a smaller inlet located on the elbow. The water exits the pipe vertically upwards from the top right. The input parameter space was chosen to be the freestream velocity at the large inlet (with a diameter of 1 m), taking values between 0.2 to 2 m s^{-1}, and the freestream velocity at the smaller inlet (with a diameter 0.5 m), taking values between 1.2 to 3 m s^{-1}. The quantities of interest were vectorised profiles at $t = 50$ s of the velocity magnitude and pressure in circular cross sections of the elbow pipe, one located at the elbow junction (oriented at 45 degrees) and the other near the pipe exit (oriented at 0 degrees). In all cases, the profiles contained 96 values. All multi-fidelity methods were applied separately to each quantity of interest, so that $d = 96$ in all cases.

The high-fidelity model (F2) was defined as LES while the low-fidelity model (F1) was defined as Sparlart-Allmaras. Both were implemented in ANSYS Fluent. The LES model employed the dynamic kinetic Energy subgrid-scale model. For the Sparlart-Allmaras model, vorticity-based production together with low-Reynold's number damping were selected. Default schemes in FLUENT were used for both F1 and F2 models, i.e., second-order implicit time stepping and central differencing for the finite-volume discretization. The meshes contained a total of 36134 nodes and 29399 hexahedral cells, and the time step was set to 0.01 s.

We assess the performance of all multi-fidelity models using a normalized root mean square error (NRMSE) for each quantity of interest. The NRMSE for N test points is defined as follows

$$\text{NRMSE} = \sqrt{\frac{\sum_{n=1}^{N}\sum_{j=1}^{d}(y_{nj} - \hat{y}_{nj})^2}{\sum_{n=1}^{N}\sum_{j=1}^{d}\hat{y}_{nj}^2}}$$

in which y_{nj} and $\hat{y}_{nj}$ denote the j-th coefficients of the n-th high-fidelity prediction and the n-th ground-truth (test) point, respectively.

We conducted tests with up to 40 F1 and 40 F2 training points, generated with randomly selected inputs. For 20 and 40 F1 training points, the number of F2 training samples was gradually increased to assess the performance of ResGP(with and without active learning) and the other methods. We computed the NRMSE against 18 F2 test points (generated in the same way as the training inputs) for each quantity of interest, averaged from a 5-fold cross validation with random shuffling of training and test data. The results are shown in Fig. 2 and Table C-1 in Appendix C, from which it is seen that ResGPgenerally outperforms NARGP and SC, especially with active learning and with a small number of F2 training data points; this is a significant advantage when data is scarce. For example, for 40 F1 and 10 F2 training points in the case of the pressure profile near the exit, ResGPwithout active learning has a 9%, 41% and 48% lower NRMSE than NARGP, SC and Greedy NAR, respectively. In the case of the velocity profile near the exit, the equivalent figures are 77%, 41% and 75%, while for the velocity profile near the pipe junction they are 51%, 80% and 97%. NARGP and ResGP without active learning converge to a similar level of error for 40 F1 training points, provided NARGP does not fail, as is the case for the velocity profile near the exit. In the latter case, the NRMSE for NARGP is around 1 for all F2 training point numbers, for both 20 and 40 F1 training points, whereas ResGPwith (without) active learning attains values of 0.1014 (0.1323) and 0.1267 (0.2039) for 20 F2 and 40 F2 training points, respectively. ResGP is more accurate with active learning, with the exception of the pressure profile near the pipe junction when the number of F2 training samples is low. For this quantity of interest, NARGP and both ResGP methods exhibit similar levels of accuracy up to 25 F2 training points for 40 F1 training samples.
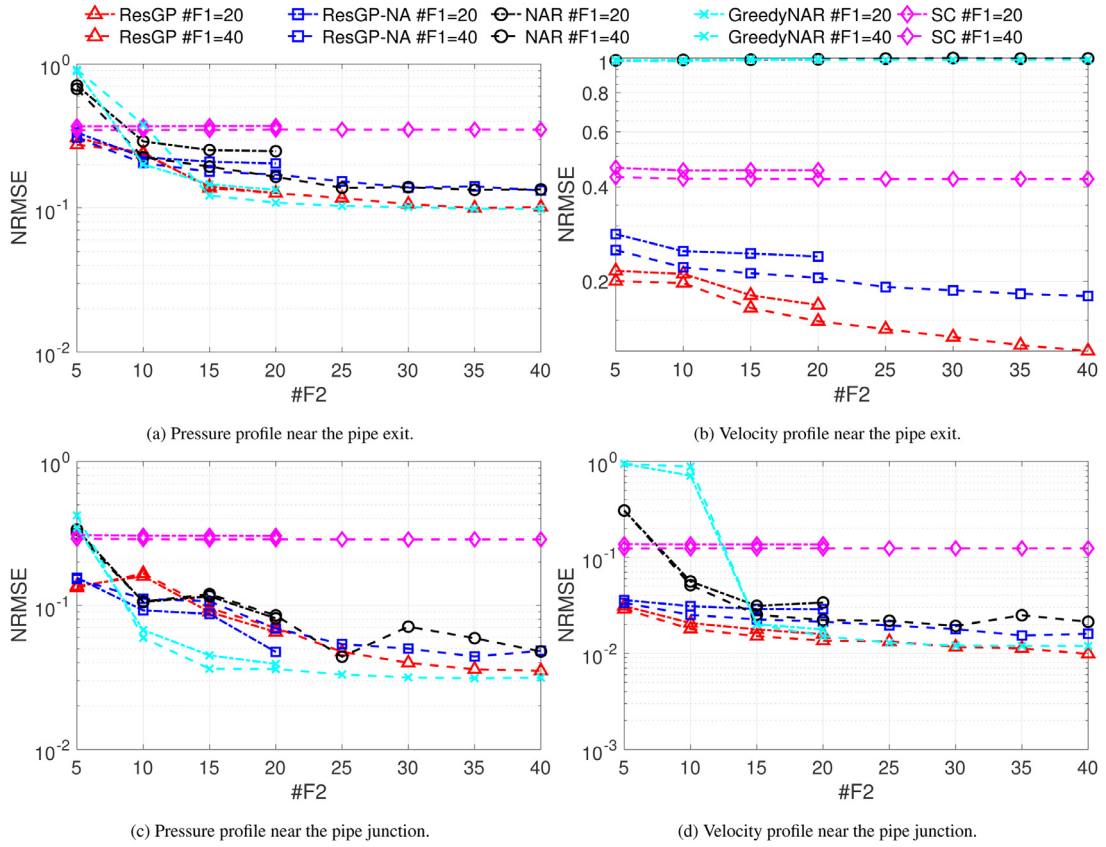
Fig. 2. Normalized root mean square errors (NRMSE) against 18 F2 test values on the two-fidelity turbulent mixing flow simulation for ResGPwith (red) and without (blue) active learning, NARGP, Greedy NAR and SC.

Greedy NAR performs well in three cases, although, like NARGP, it fails for the velocity profile near the pipe exit. For the pressure profile near the pipe junction, it exhibits the best performance, especially for a low number of F2 training points (for 40 F1 and 15 F2 training points it is at least 40% more accurate than the other methods). Noticeable, however, is that it tends to perform poorly for small numbers of F2 training points on the other quantities of interest. Also evident from these figures is that ResGPis the most robust of the methods. It is not surprising that both NARGP and Greedy NAR perform worse when F2 data is scarce, since the number of parameters in these models (Table 1) is high. Only when sufficient data is available can these parameters be learned with accuracy.

4.3. Test problem 3: molecular dynamics simulation model

In this example we consider a molecular dynamics (MD) model based on the Lennard-Jones (LJ) potential $u(\cdot)$ [39] for the interatomic interactions

$$u(r_{ij}) = 4\varepsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right],\tag{27}$$

where r_{ij} is the pairwise distance between particles i and j, ϵ is the potential well depth and σ is the length scale for the interatomic interaction. In order to prevent numerical instabilities, which can arise when the time step is too large, the magnitude of the repulsive interactions for closely approaching atoms was capped when the ratio σ/r_{ij} exceeded 1.2.

The system parameters were taken to be the temperature and density, which define the phase space. We used values of $\sigma = 3$ rA and $\varepsilon = 1$ kcal mol^{-1} to fully define the LJ potential. The domain was a cube with width $L = 27.05$ rA and periodic boundary conditions were assumed to hold on all sides. The molecular mass of each particle was set to $m = 12.01$ g mol^{-1}. The temperature T and density $\rho = Nm/V$ were used as the inputs. Here, $V = L^3$ is the box volume and N is Avogadro's number. The dimensionless density $\rho^* = N\sigma^3/V$ was therefore in the range $\rho^* \in [0.05, 0.95]$, corresponding to a molecule number in the range 36 to 696. The simulations were performed using the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) code. Integration of the equations of motion is based on the Verlet algorithm, and a micro-canonical (NVE) ensemble was used. The quantities of interest were the radial distribution function (RDF), mean squared
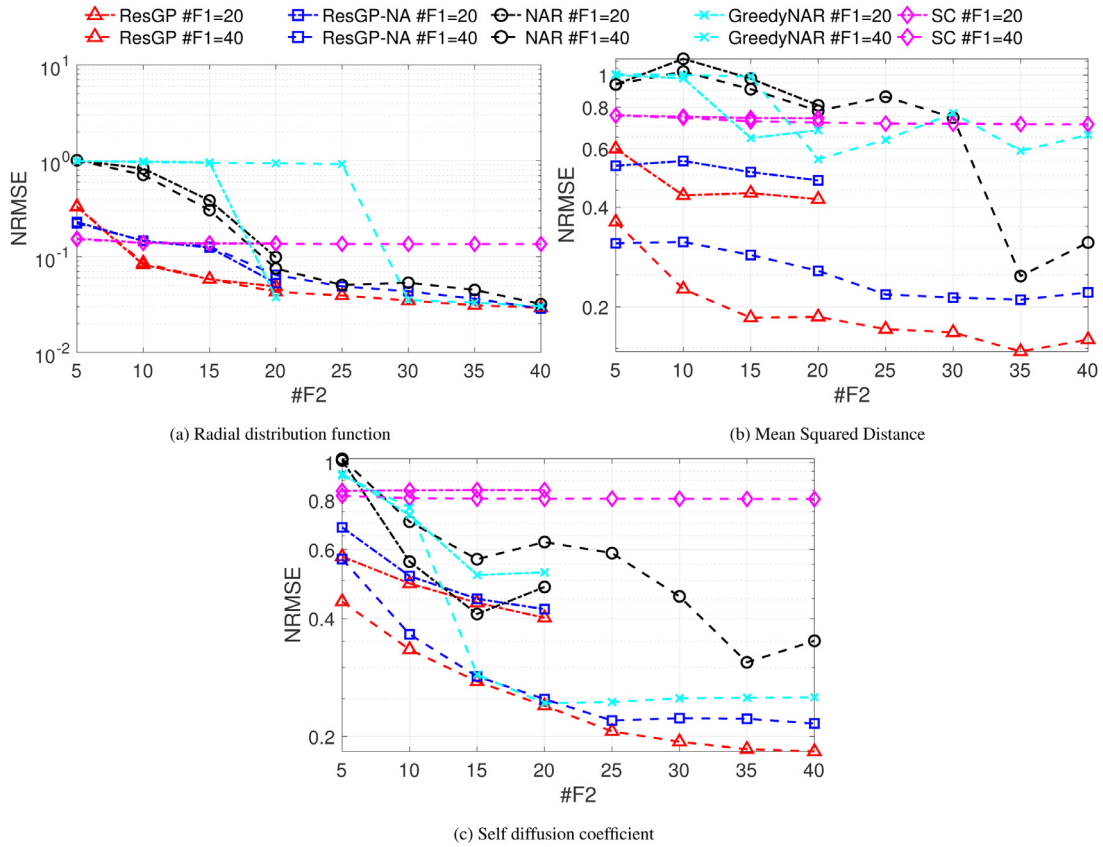
Fig. 3. Normalized root mean square errors (NRMSE) against 34 F2 test values on the two-fidelity MD simulation for ResGPwith (red) and without (blue) active learning, NARGP, Greedy NAR and SC.

displacement (MSD), and self-diffusion coefficient (SDC). All multi-fidelity methods were applied to the quantities of interest separately. The RDF was recorded at $d = 512$ points, the MSD at $d = 2000$ points and the SDC is a scalar ($d = 1$). Low-fidelity (F1) and high-fidelity (F2) simulations were defined by time steps of 10 and 1 fs, respectively. We tested all methods with up 40 F1 and 40 F2 training points, generated with randomly selected inputs in the ranges $T \in [500, 1000]$ K and $\rho \in [36.27, 701.29]$ kg m^{-3}. For 20 and 40 F1 training points, the number of F2 training points was increased gradually to assess the performance of each method by calculating the NRMSE against 34 F2 test points, generated in the same way as the training points. Again, the experiments were repeated 5 times with random shuffling of training and test data and the average NRMSE was used.

For each of the quantities of interest, the results for ResGP(with and without active learning), SC, NARGP and Greedy NAR are shown in Fig. 3 and Table C-2 in Appendix C. In this example the superiority of ResGP(both with and without active learning) is more obvious, as is the greater accuracy for low numbers of F2 training points. For example, for 40 F1 and 10 F2 training points in the case of the RDF, ResGPwith active learning has an 88%, 38% and 91% lower NRMSE than NARGP, SC and Greedy NAR, respectively. For the MSD, the equivalent figures are 78%, 70% and 77%, while for the SDC they are 53%, 59% and 57%. Active learning is seen again to enhance the performance of ResGP. For the MSD, the failure of NARGP and Greedy NAR is due to the linear scaling of the number of parameters in both methods with d (in this case $d = 2000$). For a high number of F2 training data, the accuracy of both NARGP and Greedy NAR (especially the latter) improves for the RDF and SDC. Again, this is due to the greater demand for training samples in order to learn the high number of parameters accurately.

4.4. Test problem 4: solid oxide fuel cell model

In the last example we consider a steady-state 3-d solid oxide fuel cell model. The geometry is depicted in Fig. 4. The model includes: electronic and ionic charge balances (Ohms law); the flow distribution in the gas channels (Navier-Stokes equations); flow in the porous electrodes (Brinkman equation); and gas-phase mass balances in both gas channels and the porous electrodes (Maxwell-Stefan diffusion and convection). Butler-Volmer charge transfer kinetics are assumed for the reactions in the anode ($H_2 + O^{2-} \rightarrow H_2O + 2e^-$) and cathode ($O_2 + 4e^- \rightarrow 2O^{2-}$). The cell operates in potentiostatic mode
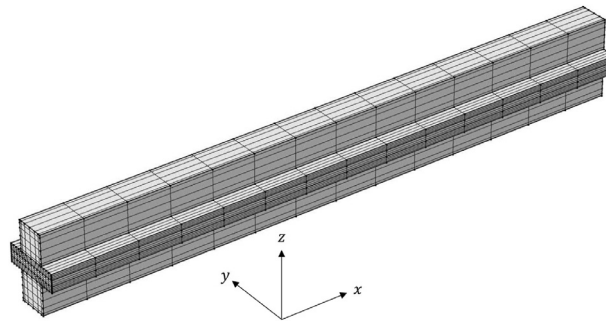
Fig. 4. Computational domain for the SOFC example: gas channels, electrodes and electrolyte, with the cathode at the top. From top to bottom, the layers are a channel, electrode, electrolyte, electrode and channel. The channel dimensions are $(x \times y \times z)$ 1 cm × 0.5 mm × 0.5 mm, the electrode dimensions are 1 cm × 1 mm × 0.1 mm, and the electrolyte dimensions are 1 cm × 1 mm × 0.1 mm. The anode inlet is located at $x = 0$ and the cathode inlet at $x = 1$ cm.
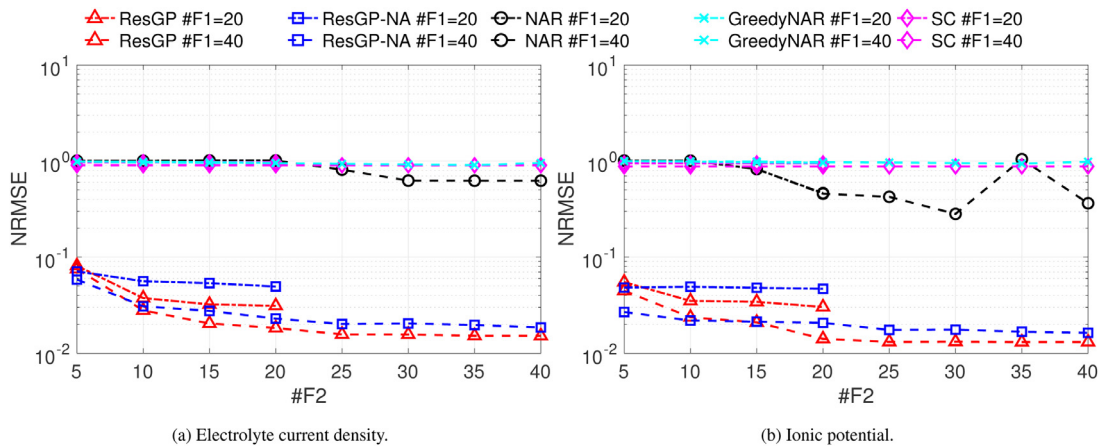

Fig. 5. Normalized root mean square errors (NRMSE) against 40 F2 test values on the two-fidelity SOFC simulation for ResGP with (red) and without (blue) active learning, NARGP, Greedy NAR and SC.

(constant cell voltage). The model was solved in COMSOL Multiphysics[2] (Application ID: 514), which is based in the finite-element method.

The inputs were taken to be the the electrode porosities $\epsilon \in [0.4, 0.85]$, the cell voltage $E_c \in [0.2, 0.85]$ V, the temperature $T \in [973, 1273]$ K, and the pressure in the channels $P \in [0.5, 2.5]$ atm. 60 inputs were selected using a Sobol sequence in the ranges indicated for the low- and high-fidelity simulations. A further 40 points were selected randomly (in the ranges above) for high-fidelity tests points. The low-fidelity F1 model used 3164 mapped elements (shown in Fig. 4) and a relative tolerance of 0.1, while the high-fidelity model used 37064 elements and a relative tolerance of 0.001. The COMSOL model also uses a V cycle geometric multigrid. The quantities of interest were taken to be profiles of the electrolyte current density (A m^{-2}) and ionic potential (V) in the $x - z$ plane located at the centre of the channels (Fig. 4). In both cases, the number of points recorded was $d = 100 \times 50 = 5000$ and both profiles were vectorised to form the training and test outputs.

The NRMSE (with five-fold cross validation) for 20 and 40 F1 training samples and an increasing number of F2 training samples is shown in Fig. 5 and Table C-3 in Appendix C for ResGP(with and without active learning), NARGP, Greedy NAR and SC. In this example, none of the other methods worked well, which is hardly surprising in the case of NARGP and Greedy NAR considering that $d = 5000$. For SC, without out-of-sample F1 data, the performance is generally poor. ResGP, on the other hand, shows a steady decline in the NRMSE as the number of F2 training samples is increased. For 40 F1 and 10 F2 training points, ResGP with active learning has a 97% lower NRMSE than each of the other methods for the electrolyte current density prediction. For the ionic potential, the equivalent values are almost identical, and these levels of improvement in the accuracy are maintained up to 40 F2 training points (over 90% lower NRSME than all of the other methods).

Predictions of the quantities of interest for 40 F1 and 20 F2 training points using ResGP(without active learning) are shown in Figs. 6 and 7, together with the ground truths (tests) and pointwise absolute differences. The electrolyte current density predictions in Fig. 6 correspond to the lowest error ($\epsilon = 0.67$, $E_c = 0.32$ V, $T = 1151.2$ K, $P = 1.63$ atm), the median error ($\epsilon = 0.48$, $E_c = 0.36$ V, $T = 981.7$ K, $P = 1.97$ atm) and the highest error ($\epsilon = 0.75$, $E_c = 0.74$ V, $T = 1025.7$ K, $P = 1.24$
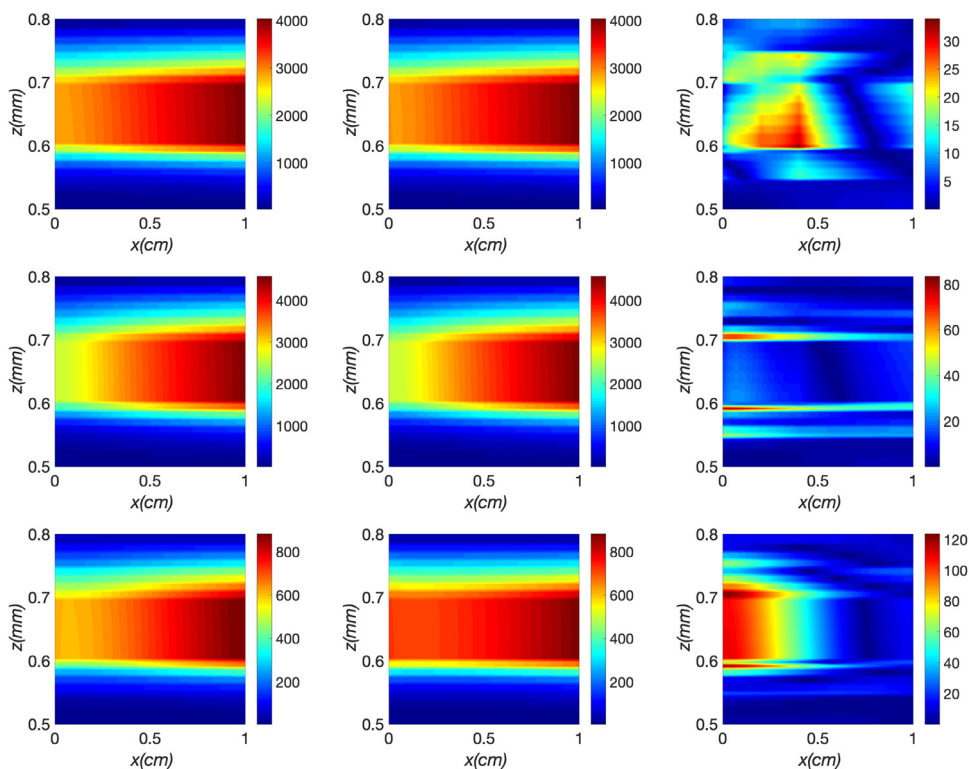
Fig. 6. Predictions of the electrolyte current density (A m^{-2}) in the $x-z$ plane located at the centre of the channels in Fig. 4 for 40 F1 and 20 F2 training points. These predictions correspond from the top to bottom row to the lowest error, the median error and the highest error for the 5-fold cross validation. The columns from left to right are the prediction using ResGP, the ground truth (test) and the pointwise absolute differences.

Fig. 7. Predictions of the ionic potential (V) in the $x-z$ plane located at the centre of the channels in Fig. 4 for 40 F1 and 20 F2 training points. These predictions correspond from the top to bottom row to the lowest error, the median error and the highest error for the 5-fold cross validation. The columns from left to right are the prediction using ResGP, the ground truth (test) and the pointwise absolute differences.
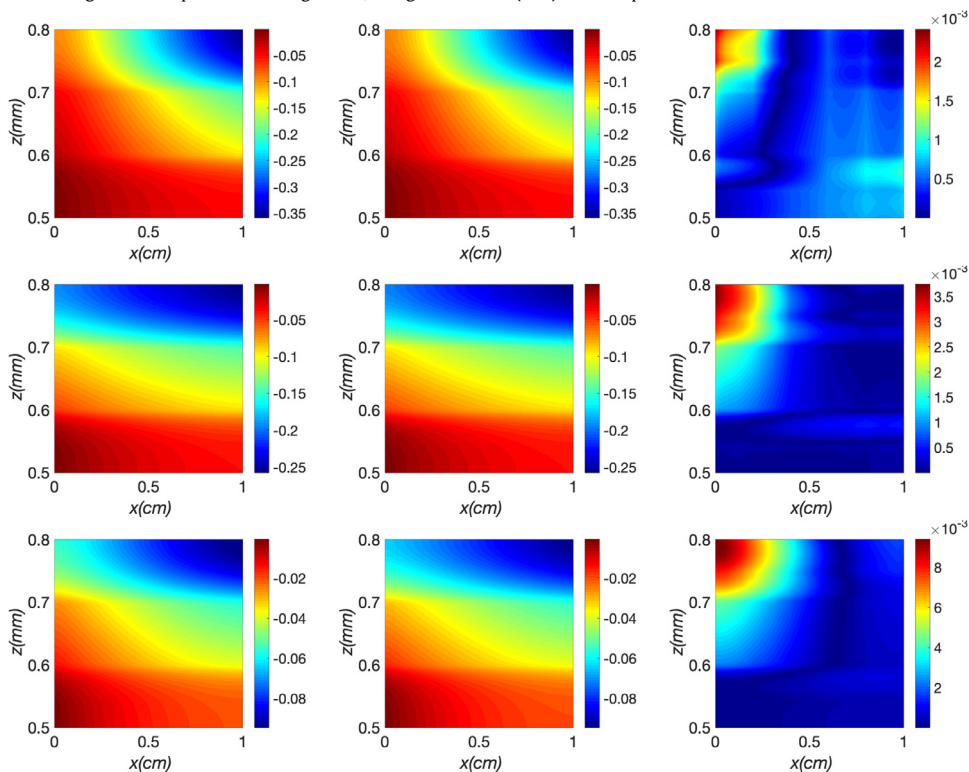
atm) for the 5-fold cross validation at 40 F1 and 20 F2 training points in Fig. 5(a). Likewise, the ionic potential predictions in Fig. 7 correspond to the lowest error ($\epsilon = 0.79$, $E_c = 0.28$ V, $T = 1191.4$ K, $P = 2.12$ atm), the median error ($\epsilon = 0.59$, $E_c = 0.30$ V, $T = 1080.7$ K, $P = 1.87$ atm) and the highest error ($\epsilon = 0.81$, $E_c = 0.69$ V, $T = 926.6$ K, $P = 1.04$ atm) for the 5-fold cross validation at 40 F1 and 20 F2 training points in Fig. 5(b). Even in the case of the largest error, the qualitative and quantitative accuracy of ResGP is high.

5. Summary and conclusions

In this paper we introduced an additive residual structure for multi-fidelity models in order to capture the connection between data at different fidelities. The result is a non-parametric Bayesian model that is equipped with a closed-form solution for the predictive posterior. This permits tasks such as uncertainty estimation and Bayesian optimization to be conducted efficiently and accurately. Under a noise-free assumption for the multi-fidelity data, the model scales efficiently accurately to high-dimensional problems. We derive error bounds for the univariate case, which may be of use in applications such as control.

Four benchmark problems in a variety of settings demonstrated that ResGPcan not only provide accurate posterior predictions but also faithful estimates of model uncertainty. The great drawback of SC is the requirement of out-of-sample simulations for making predictions, which the other methods, including ResGP, avoid. When comparing with NARGP and Greedy NAR, the advantages of ResGPare clear, not least in terms of stable and accurate predictions for high-dimensional problems. Both of these methods require large numbers of high-fidelity training points to yield accurate predictions, since the number of parameters is high. This is a major drawback and a significant advantage of ResGP, which is particularly good for sparse high-fidelity data. Moreover, the accuracy is further improved for small data sets by appealing to active learning *via* variance reduction. Lastly, the performance of ResGPis found to be markedly superior for high-dimensional problems since the number of parameters does not scale with the output space dimensionality. This makes ResGPapplicable to a broader range of problems than other state-of-the-art methods.

Acknowledgments

Appendix A. Proofs

Proof Lemma 3.3. Using the definition of $\mu^F(\xi)$ in (8) for the univariate case, we have

$$
\begin{aligned}
\left| \mu^F(\xi) - \mu^F(\xi') \right| &= \left| \sum_{f=1}^{F} \left(\mu_r^f(\xi) - \mu_r^f(\xi') \right) \right| \\
&\leq \sum_{f=1}^{F} \left| \mu_r^f(\xi) - \mu_r^f(\xi') \right| \\
&= \sum_{f=1}^{F} \left| \left[\mathbf{k}^f(\xi) - \mathbf{k}^f(\xi') \right]^T (\mathbf{k}^f)^{-1} \mathbf{r}^f \right| \\
&\leq \sum_{f=1}^{F} L_k^f \sqrt{N_f} \|\xi - \xi'\| \|(\mathbf{k}^f)^{-1} \mathbf{r}^f\| \\
&= \left(\sum_{f=1}^{F} L_k^f \sqrt{N_f} \|(\mathbf{k}^f)^{-1} \mathbf{r}^f\| \right) \|\xi - \xi'\|, \quad \forall \xi, \xi' \in \mathcal{X},
\end{aligned}
\tag{A-1}
$$

from which the bound on the Lipschitz constant L_{μ^F} is derived.

For the standard deviation bound, we use the definitions of $v_r^f(\boldsymbol{\xi})$, the univariate equivalent of $\mathbf{V}_r^f(\boldsymbol{\xi})$, and $v^F(\boldsymbol{\xi})$, the equivalent of $\mathbf{V}^F(\boldsymbol{\xi})$, in (11), to obtain

$$
\begin{aligned}
\left| v^F(\boldsymbol{\xi}) - v^F(\boldsymbol{\xi}') \right| & \leq \sum_{f=1}^{F} \left| v_r^f(\boldsymbol{\xi}) - v_r^f(\boldsymbol{\xi}') \right| \\
& = \sum_{f=1}^{F} \left| k^f(\boldsymbol{\xi}, \boldsymbol{\xi}|\boldsymbol{\theta}^f) - k^f(\boldsymbol{\xi}, \boldsymbol{\xi}'|\boldsymbol{\theta}^f) + k^f(\boldsymbol{\xi}, \boldsymbol{\xi}'|\boldsymbol{\theta}^f) - k^f(\boldsymbol{\xi}', \boldsymbol{\xi}'|\boldsymbol{\theta}^f) \right| \\
& \leq \sum_{f=1}^{F} 2L_k^f \|\boldsymbol{\xi} - \boldsymbol{\xi}'\| + \|\mathbf{k}^f(\boldsymbol{\xi}') - \mathbf{k}^f(\boldsymbol{\xi})\| \|(\mathbf{k}^f)^{-1}\|_2 \|\mathbf{k}^f(\boldsymbol{\xi}) + \mathbf{k}^f(\boldsymbol{\xi}')\|, \quad \forall \boldsymbol{\xi}, \boldsymbol{\xi}' \in \mathcal{X}.
\end{aligned}
\tag{A-2}
$$

Since

$$
\begin{aligned}
\|\mathbf{k}^f(\boldsymbol{\xi}') - \mathbf{k}^f(\boldsymbol{\xi})\| & \leq \sqrt{N_f} L_k^f \|\boldsymbol{\xi} - \boldsymbol{\xi}'\| \\
\|\mathbf{k}^f(\boldsymbol{\xi}) + \mathbf{k}^f(\boldsymbol{\xi}')\| & \leq 2\sqrt{N_f} \max_{\boldsymbol{\xi}, \boldsymbol{\xi}' \in \mathcal{X}} k^f(\boldsymbol{\xi}, \boldsymbol{\xi}'|\boldsymbol{\theta}^f),
\end{aligned}
\tag{A-3}
$$

we obtain

$$
\left| v^F(\boldsymbol{\xi}) - v^F(\boldsymbol{\xi}') \right| \leq \sum_{f=1}^{F} 2L_k^f \|\boldsymbol{\xi} - \boldsymbol{\xi}'\| + \|\mathbf{k}^f(\boldsymbol{\xi}') - \mathbf{k}^f(\boldsymbol{\xi})\| \|(\mathbf{k}^f)^{-1}\|_2 \|\mathbf{k}^f(\boldsymbol{\xi}) + \mathbf{k}^f(\boldsymbol{\xi}')\|,
\tag{A-4}
$$

To obtain the modulus of continuity for the standard deviation $\sigma^F(\boldsymbol{\xi}) = \sqrt{v^F(\boldsymbol{\xi})} = \sqrt{\sum_{f=1}^{F} v_r^f(\boldsymbol{\xi})}$ we first write

$$
\left| v^F(\boldsymbol{\xi}) - v^F(\boldsymbol{\xi}') \right| = \left| \sigma^F(\boldsymbol{\xi}) - \sigma^F(\boldsymbol{\xi}') \right| \left| \sigma^F(\boldsymbol{\xi}) + \sigma^F(\boldsymbol{\xi}') \right| \geq \left| \sigma^F(\boldsymbol{\xi}) - \sigma^F(\boldsymbol{\xi}') \right|^2,
\tag{A-5}
$$

since $\left| \sigma^F(\boldsymbol{\xi}) + \sigma^F(\boldsymbol{\xi}') \right| \geq \left| \sigma^F(\boldsymbol{\xi}) - \sigma^F(\boldsymbol{\xi}') \right|$ by the positive semidefiniteness of the standard deviation. This yields

$$
\left| \sigma^F(\boldsymbol{\xi}) - \sigma^F(\boldsymbol{\xi}') \right| \leq \left[\sum_{f=1}^{F} 2L_k^f \left(1 + N_f \max_{\boldsymbol{\xi}, \boldsymbol{\xi}' \in \mathcal{X}} k^f(\boldsymbol{\xi}, \boldsymbol{\xi}'|\boldsymbol{\theta}^f) \|(\mathbf{k}^f)^{-1}\|_2 \right) \|\boldsymbol{\xi} - \boldsymbol{\xi}'\| \right]^{1/2}, \quad \forall \boldsymbol{\xi}, \boldsymbol{\xi}' \in \mathcal{X},
\tag{A-6}
$$

from which we obtain $\omega_\sigma^F(\cdot)$.   □

Proof Theorem 3.5. Pick $\delta \in (0, 1)$. For every design $\mathcal{X}_\tau^F \subset \mathcal{X}$ with $|\mathcal{X}_\tau^F|$ grid points and for a fill distance for this design satisfying

$$
\sup_{\boldsymbol{\xi} \in \mathcal{X}} \min_{\boldsymbol{\xi}' \in \mathcal{X}_\tau^F} \|\boldsymbol{\xi} - \boldsymbol{\xi}'\| \leq \tau
\tag{A-7}
$$

it holds with probability of at least $1 - |\mathcal{X}_\tau^F| e^{-\beta(\tau)/2}$ that [31] [Lemma 5.1]

$$
|y^F(\boldsymbol{\xi}) - \mu^F(\boldsymbol{\xi})| \leq \sqrt{\beta(\tau)} \sigma^F(\boldsymbol{\xi}), \quad \forall \boldsymbol{\xi} \in \mathcal{X}_\tau^F.
\tag{A-8}
$$

Choosing $\beta(\tau) = 2 \log \left(\frac{|\mathcal{X}_\tau^F|}{\delta} \right)$, the inequality

$$
|y^F(\boldsymbol{\xi}) - \mu^F(\boldsymbol{\xi})| \leq \sqrt{\beta(\tau)} \sigma^F(\boldsymbol{\xi}), \quad \forall \boldsymbol{\xi} \in \mathcal{X}_\tau^F
\tag{A-9}
$$

holds with probability of at least $1 - \delta$. Using Lemma 3.3 and the continuity of $y^F(\boldsymbol{\xi})$, as well as the monotonicity of $\omega_\sigma^F(\cdot)$, for every $\boldsymbol{\xi}' \in \mathcal{X}_\tau^F$ and $\boldsymbol{\xi} \in \mathcal{X}$ we obtain

$$
\begin{aligned}
|y^F(\boldsymbol{\xi}) - \mu^F(\boldsymbol{\xi})| & \leq |y^F(\boldsymbol{\xi}) - y^F(\boldsymbol{\xi}')| + |\mu^F(\boldsymbol{\xi}') - \mu^F(\boldsymbol{\xi})| + |y^F(\boldsymbol{\xi}') - \mu^F(\boldsymbol{\xi}')| \\
& \leq L_y \|\boldsymbol{\xi} - \boldsymbol{\xi}'\| + L_{\mu^F} \|\boldsymbol{\xi} - \boldsymbol{\xi}'\| + \sqrt{\beta(\tau)} \sigma^F(\boldsymbol{\xi}') \\
& \leq \tau L_y + \tau L_{\mu^F} + \sqrt{\beta(\tau)} \sigma^F(\boldsymbol{\xi}') \\
& = \tau L_y + \tau L_{\mu^F} + \sqrt{\beta(\tau)} |\sigma^F(\boldsymbol{\xi}) + \sigma^F(\boldsymbol{\xi}') - \sigma^F(\boldsymbol{\xi})| \\
& \leq \tau L_y + \tau L_{\mu^F} + \sqrt{\beta(\tau)} \omega_\sigma^F(\tau) + \sqrt{\beta(\tau)} \sigma^F(\boldsymbol{\xi}).
\end{aligned}
\tag{A-10}
$$

The final result follows from the fact that the minimum number of grid points satisfying condition (A-7) is given by $M(\tau, \mathcal{X})$.   □

Appendix B. Synthetic examples

1. The Currin function is a two-dimensional problem with inputs $\boldsymbol{\xi} \in [0, 1]^2$. The high- and low-delity functions are given respectively by

$$y^H(\boldsymbol{\xi}) = \left[1 - \exp\left(-\frac{1}{2\xi_2}\right)\right] \frac{2300\xi_1^3 + 1900\xi_1^2 + 2092\xi_1 + 60}{100\xi_1^3 + 500\xi_1^2 + 4\xi_1 + 20}, \tag{B-1}$$

and

$$y^L(\boldsymbol{\xi}) = \frac{1}{4}\left[y^H(\xi_1 + 0.05, \xi_2 + 0.05) + y^H(\xi_1 + 0.05, \max(0, \xi_2 - 0.05))\right] + \tag{B-2}$$

$$\frac{1}{4}\left[y^H(\xi_1 - 0.05, \xi_2 + 0.05) + y^H(\xi_1 - 0.05, \max(0, \xi_2 - 0.05))\right]. \tag{B-3}$$

2. The Park function is a four-dimensional problem with inputs $\boldsymbol{\xi} \in [0, 1]^4$. The high- and low-delity functions are given respectively by

$$y^H(\boldsymbol{\xi}) = \frac{\xi_1}{2}\left[\sqrt{1 + (\xi_2 + \xi_3^2)\frac{\xi_4}{\xi_1^2}} - 1\right] + (\xi_1 + 3\xi_4)\exp\left[1 + \sin(\xi_3)\right]. \tag{B-4}$$

$$y^L(\boldsymbol{\xi}) = \left[1 + \frac{\sin(\xi_1)}{10}\right]y^H(\boldsymbol{\xi}) - 2\xi_1 + \xi_2^2 + \xi_3^2 + 0.5. \tag{B-5}$$

3. The Borehole example is an eight-dimensional problem with inputs $\xi_1 \in [0.05, 0.15]$, $\xi_2 \in [100, 50000]$, $\xi_3 \in [63070, 115600]$, $\xi_4 \in [990, 1110]$, $\xi_5 \in [63.1, 115]$, $\xi_6 \in [700, 820]$, $\xi_7 \in [1120, 1680]$, $\xi_8 \in [9855, 12045]$. The high- and low-delity functions are given respectively by

$$y^H(\boldsymbol{\xi}) = \frac{2\pi\xi_3(\xi_4 - \xi_6)}{\log(\xi_2/\xi_1)\left(1 + \frac{2\xi_7\xi_3}{\log(\xi_2/\xi_1)\xi_1^2\xi_8}\right) + \frac{\xi_3}{\xi_5}}, \tag{B-6}$$

$$y^L(\boldsymbol{\xi}) = \frac{5\xi_3(\xi_4 - \xi_6)}{\log(\xi_2/\xi_1)\left(1.5 + \frac{2\xi_7\xi_3}{\log(\xi_2/\xi_1)\xi_1^2\xi_8}\right) + \frac{\xi_3}{\xi_5}}. \tag{B-7}$$

4. The three-level Branin function is a two-dimensional problem with inputs $[-5, 10] \times [0, 15]$. Three fidelities are considered, defined by

$$y^1(\boldsymbol{\xi}) = \left(\frac{-1.275\xi_1^2}{\pi^2} + \frac{5\xi_1}{\pi} + \xi_2 - 6\right)^2 + \left(10 - \frac{5}{4\pi}\right)\cos(\xi_1) + 10, \tag{B-8}$$

$$y^2(\boldsymbol{\xi}) = 10\sqrt{y^H(\boldsymbol{\xi} - 2)} + 2(\xi_1 - 0.5) - 3(3\xi_2 - 1) - 1, \tag{B-9}$$

$$y^3(\boldsymbol{\xi}) = y^2\big(1.2(\boldsymbol{\xi} + 2)\big) - 3\xi_2 + 1. \tag{B-10}$$

5. The Hartmann-3d example has inputs in $[0, 1]^3$. The fidelity $f = 1, 2, 3$ observations are given by

$$y^f(\boldsymbol{\xi}) = \sum_{i=1}^4 \alpha_i \exp\left(-\sum_{j=1}^3 A_{ij}(\xi_j - P_{ij})^2\right), \tag{B-11}$$

where

$$A = \begin{bmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} 0.3689 & 0.1170 & 0.2673 \\ 0.4699 & 0.4387 & 0.7470 \\ 0.1091 & 0.8732 & 0.5547 \\ 0.0381 & 0.5743 & 0.8828 \end{bmatrix}.$$

$\boldsymbol{\alpha}$ is set to $(1.0, 1.2, 3.0, 3.2)^T$ and is updated to $\boldsymbol{\alpha}_f = \boldsymbol{\alpha} + (3 - f)\boldsymbol{\delta}$ for lower fidelities, in which $\boldsymbol{\delta} = (0.01, -0.01, -0.1, 0.1)^T$.

Appendix C. Normalized root mean square errors on the test sets for the multivariate examples 4.1-4.3

Table C-1

Normalized root mean square errors (NRMSE) against 18 F2 test values on the two-fidelity turbulent mixing flow simulation for ResGPwith (ResGP) and without (ResGP-NA) active learning, NARGP, Greedy NAR and SC (see Fig. 2).

	N_1	$N_2 = 5$	$N_2 = 10$	$N_2 = 15$	$N_2 = 20$	$N_2 = 25$	$N_2 = 30$	$N_2 = 35$	$N_2 = 40$
				Pressure profile near the pipe exit					
ResGP	20	0.3080	0.2369	0.1409	0.1267	N/A	N/A	N/A	N/A
ResGP	40	0.2750	0.2428	0.1374	0.1267	0.1171	0.1063	0.0999	0.1014
ResGP-NA	20	0.3344	0.2250	0.2098	0.2039	N/A	N/A	N/A	N/A
ResGP-NA	40	0.3074	0.2051	0.1788	0.1706	0.1531	0.1387	0.1412	0.1323
NARGP	20	0.7096	0.2902	0.2524	0.2481	N/A	N/A	N/A	N/A
NARGP	40	0.6716	0.2253	0.1939	0.1645	0.1380	0.1395	0.1337	0.1347
SC	20	0.3694	0.3694	0.3706	0.3706	N/A	N/A	N/A	N/A
SC	40	0.3474	0.3486	0.3484	0.3507	0.3502	0.3502	0.3504	0.3504
GreedyNAR	20	0.8927	0.2006	0.1461	0.1335	N/A	N/A	N/A	N/A
GreedyNAR	40	0.9117	0.3801	0.1222	0.1088	0.1032	0.1014	0.0989	0.0985
				Velocity profile near the pipe exit					
ResGP	20	0.2159	0.2111	0.1803	0.1679	N/A	N/A	N/A	N/A
ResGP	40	0.2002	0.1973	0.1643	0.1491	0.1406	0.1327	0.1250	0.1197
ResGP-NA	20	0.2826	0.2495	0.2453	0.2399	N/A	N/A	N/A	N/A
ResGP-NA	40	0.2515	0.2215	0.2122	0.2051	0.1917	0.1871	0.1823	0.1794
NARGP	20	1.0145	1.0171	1.0188	1.0221	N/A	N/A	N/A	N/A
NARGP	40	1.0145	1.0171	1.0188	1.0221	1.0275	1.0320	1.0279	1.0287
SC	20	0.4606	0.4514	0.4521	0.4522	N/A	N/A	N/A	N/A
SC	40	0.4297	0.4248	0.4246	0.4237	0.4240	0.4241	0.4239	0.4239
GreedyNAR	20	1.0165	1.0148	1.0231	1.0232	N/A	N/A	N/A	N/A
GreedyNAR	40	1.0102	1.0085	1.0178	1.0181	1.0175	1.0181	1.0178	1.0198
				Pressure profile near the pipe junction					
ResGP	20	0.1363	0.1593	0.0898	0.0653	N/A	N/A	N/A	N/A
ResGP	40	0.1326	0.1665	0.0955	0.0697	0.0475	0.0398	0.0360	0.0351
ResGP-NA	20	0.1556	0.0923	0.0872	0.0475	N/A	N/A	N/A	N/A
ResGP-NA	40	0.1523	0.1111	0.1067	0.0690	0.0538	0.0500	0.0441	0.0483
NARGP	20	0.3360	0.1055	0.1156	0.0812	N/A	N/A	N/A	N/A
NARGP	40	0.3153	0.1061	0.1197	0.0851	0.0440	0.0711	0.0591	0.0478
SC	20	0.3088	0.3048	0.3045	0.3041	N/A	N/A	N/A	N/A
SC	40	0.2904	0.2884	0.2873	0.2879	0.2874	0.2870	0.2876	0.2874
GreedyNAR	20	0.5140	0.0602	0.0434	0.0371	N/A	N/A	N/A	N/A
GreedyNAR	40	0.6438	0.0610	0.0463	0.0321	0.0308	0.0293	0.0346	0.0345
				Velocity profile near the pipe junction					
ResGP	20	0.0312	0.0207	0.0177	0.0158	N/A	N/A	N/A	N/A
ResGP	40	0.0289	0.0180	0.0151	0.0136	0.0133	0.0116	0.0113	0.0099
ResGP-NA	20	0.0359	0.0310	0.0290	0.0288	N/A	N/A	N/A	N/A
ResGP-NA	40	0.0336	0.0252	0.0225	0.0214	0.0196	0.0179	0.0153	0.0160
NARGP	20	0.3080	0.0561	0.0312	0.0338	N/A	N/A	N/A	N/A
NARGP	40	0.3076	0.0514	0.0252	0.0222	0.0218	0.0193	0.0249	0.0213
SC	20	0.1373	0.1367	0.1367	0.1367	N/A	N/A	N/A	N/A
SC	40	0.1238	0.1241	0.1241	0.1242	0.1242	0.1242	0.1242	0.1242
GreedyNAR	20	0.9421	0.7107	0.0201	0.0177	N/A	N/A	N/A	N/A
GreedyNAR	40	0.9456	0.8824	0.0199	0.0151	0.0129	0.0120	0.0119	0.0119

Table C-2

Normalized root mean square errors (NRMSE) against 34 F2 test values on the two-fidelity MD simulation for ResGP with (ResGP) and without (ResGP-NA) active learning, NARGP, Greedy NAR and SC (see Fig. 3).

	N_1	$N_2 = 5$	$N_2 = 10$	$N_2 = 15$	$N_2 = 20$	$N_2 = 25$	$N_2 = 30$	$N_2 = 35$	$N_2 = 40$
				Radial distribution function					
ResGP	20	0.3296	0.0823	0.0580	0.0487	N/A	N/A	N/A	N/A
ResGP	40	0.3318	0.0860	0.0583	0.0428	0.0392	0.0348	0.0310	0.0293
ResGP-NA	20	0.2290	0.1454	0.1244	0.0527	N/A	N/A	N/A	N/A
ResGP-NA	40	0.2222	0.1466	0.1269	0.0644	0.0488	0.0434	0.0363	0.0287
NARGP	20	1.0103	0.8290	0.3844	0.0988	N/A	N/A	N/A	N/A
NARGP	40	1.0103	0.7109	0.3046	0.0751	0.0505	0.0534	0.0449	0.0319
SC	20	0.1540	0.1393	0.1374	0.1372	N/A	N/A	N/A	N/A
SC	40	0.1518	0.1385	0.1367	0.1357	0.1354	0.1352	0.1351	0.1351
GreedyNAR	20	0.9900	0.9806	0.9565	0.0376	N/A	N/A	N/A	N/A
GreedyNAR	40	0.9952	0.9658	0.9504	0.9406	0.9218	0.0351	0.0331	0.0303
				Mean Squared Distance					
ResGP	20	0.5999	0.4338	0.4408	0.4223	N/A	N/A	N/A	N/A
ResGP	40	0.3610	0.2264	0.1855	0.1866	0.1713	0.1675	0.1467	0.1593
ResGP-NA	20	0.5331	0.5510	0.5101	0.4814	N/A	N/A	N/A	N/A
ResGP-NA	40	0.3111	0.3143	0.2869	0.2570	0.2180	0.2135	0.2102	0.2213
NARGP	20	0.9370	1.1201	0.9763	0.8112	N/A	N/A	N/A	N/A
NARGP	40	0.9383	1.0250	0.9072	0.7806	0.8609	0.7446	0.2478	0.3128
SC	20	0.7563	0.7500	0.7422	0.7417	N/A	N/A	N/A	N/A
SC	40	0.7554	0.7434	0.7258	0.7200	0.7145	0.7139	0.7114	0.7107
GreedyNAR	20	1.0001	0.9783	0.6462	0.6822	N/A	N/A	N/A	N/A
GreedyNAR	40	1.0062	1.0007	0.9903	0.5576	0.6384	0.7679	0.5913	0.6605
				Self diffusion coefficient					
ResGP	20	0.5757	0.4915	0.4389	0.4017	N/A	N/A	N/A	N/A
ResGP	40	0.4418	0.3335	0.2765	0.2397	0.2062	0.1940	0.1857	0.1833
ResGP-NA	20	0.6836	0.5135	0.4491	0.4226	N/A	N/A	N/A	N/A
ResGP-NA	40	0.5674	0.3646	0.2848	0.2492	0.2198	0.2228	0.2221	0.2160
NARGP	20	1.0133	0.5587	0.4109	0.4816	N/A	N/A	N/A	N/A
NARGP	40	1.0226	0.7069	0.5668	0.6271	0.5875	0.4556	0.3093	0.3510
SC	20	0.8465	0.8497	0.8512	0.8506	N/A	N/A	N/A	N/A
SC	40	0.8227	0.8109	0.8089	0.8083	0.8086	0.8080	0.8074	0.8072
GreedyNAR	20	0.9353	0.7357	0.5170	0.5246	N/A	N/A	N/A	N/A
GreedyNAR	40	0.9277	0.7682	0.2876	0.2433	0.2450	0.2504	0.2513	0.2520

Table C-3

Normalized root mean square errors (NRMSE) against 40 F2 test values on the two-fidelity SOFC simulation for ResGP with (ResGP) and without (ResGP-NA) active learning, NARGP, Greedy NAR and SC (see Fig. 5).

	N_1	$N_2 = 5$	$N_2 = 10$	$N_2 = 15$	$N_2 = 20$	$N_2 = 25$	$N_2 = 30$	$N_2 = 35$	$N_2 = 40$
				Electrolyte current density					
ResGP	20	0.0811	0.0375	0.0322	0.0311	N/A	N/A	N/A	N/A
ResGP	40	0.0752	0.0279	0.0204	0.0183	0.0157	0.0157	0.0152	0.0151
ResGP-NA	20	0.0711	0.0561	0.0536	0.0494	N/A	N/A	N/A	N/A
ResGP-NA	40	0.0585	0.0308	0.0275	0.0229	0.0201	0.0204	0.0196	0.0185
NARGP	20	1.0079	1.0063	1.0142	1.0138	N/A	N/A	N/A	N/A
NARGP	40	1.0079	1.0063	1.0142	1.0138	0.8145	0.6257	0.6245	0.6247
SC	20	0.9745	0.9752	0.9752	0.9752	N/A	N/A	N/A	N/A
SC	40	0.9048	0.9053	0.9054	0.9054	0.9054	0.9054	0.9054	0.9054
GreedyNAR	20	0.9895	0.9758	0.9641	0.9524	N/A	N/A	N/A	N/A
GreedyNAR	40	0.9890	0.9778	0.9664	0.9527	0.9409	0.9274	0.9127	0.9359
				Ionic potential					
ResGP	20	0.0549	0.0350	0.0342	0.0302	N/A	N/A	N/A	N/A
ResGP	40	0.0446	0.0236	0.0209	0.0141	0.0131	0.0132	0.0130	0.0131
ResGP-NA	20	0.0482	0.0491	0.0479	0.0467	N/A	N/A	N/A	N/A
ResGP-NA	40	0.0268	0.0219	0.0213	0.0207	0.0175	0.0176	0.0167	0.0163
NARGP	20	1.0106	1.0078	0.8257	0.4638	N/A	N/A	N/A	N/A
NARGP	40	1.0106	1.0078	0.8252	0.4573	0.4243	0.2819	1.0458	0.3637
SC	20	0.9460	0.9464	0.9465	0.9465	N/A	N/A	N/A	N/A
SC	40	0.8798	0.8800	0.8800	0.8800	0.8800	0.8800	0.8800	0.8800
GreedyNAR	20	0.9974	0.9861	0.9730	0.9586	N/A	N/A	N/A	N/A
GreedyNAR	40	1.0038	0.9970	0.9905	0.9745	0.9661	0.9532	0.9399	0.9843

References

[1] B. Peherstorfer, K. Willcox, M. Gunzburger, Survey of multifidelity methods in uncertainty propagation, inference, and optimization, SIAM Rev. 60 (3) (2018) 550–591.
[2] M.C. Kennedy, A. O'Hagan, Predicting the output from a complex computer code when fast approximations are available, Biometrika 87 (1) (2000) 1–13.
[3] F.A.C. Viana, T.W. Simpson, V. Balabanov, V. Toropov, Special section on multidisciplinary design optimization: metamodeling in multidisciplinary design optimization: how far have we really come? AIAA J. 52 (4) (2014) 670–690.
[4] S. Galelli, A. Castelletti, A. Goedbloed, High-performance integrated control of water quality and quantity in urban water reservoirs, Water Resour. Res. 51 (11) (2015) 9053–9072.
[5] T. Santner, B. Williams, W. Notz, The Design and Analysis of Computer Experiments, Springer, 2003.
[6] N.E. Owen, P. Challenor, P.P. Menon, S. Bennani, Comparison of surrogate-based uncertainty quantification methods for computationally expensive simulators, SIAM/ASA J. Uncertain. Quantif. 5 (1) (2017) 403–435.
[7] S. Conti, A. O'Hagan, Bayesian emulation of complex multi-output and dynamic computer models, J. Stat. Plan. Inference 140 (2010) 640–651.
[8] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[9] G. Venter, R.T. Haftka, J.H. Starnes, Construction of response surface approximations for design optimization, AIAA J. 36 (12) (1998) 2242–2249.
[10] M.D. Gunzburger, J.S. Peterson, J.N. Shadid, Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data, Comput. Methods Appl. Mech. Eng. 196 (2007) 1030–1047.
[11] M.G. Fernández-Godino, C. Park, N.-H. Kim, R.T. Haftka. Review of multi-fidelity models. (2016) arXiv preprint arXiv:1609.07196.
[12] L. Leifsson, S. Koziel, Aerodynamic shape optimization by variable-fidelity computational fluid dynamics models: a review of recent progress, J. Comput. Sci. 10 (2015) 45–54.
[13] L. Le Gratiet, Multi-fidelity Gaussian process regression for computer experiments, Université Paris-Diderot-Paris VII, 2013 Ph.D. thesis.
[14] P. Perdikaris, M. Raissi, A. Damianou, N. Lawrence, G.E. Karniadakis, Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling, Proc. R. Soc. A Math. Phys. Eng. Sci. 473 (2198) (2017) 20160751.
[15] L. Parussini, D. Venturi, P. Perdikaris, G. Karniadakis, Multi-fidelity Gaussian process regression for prediction of random fields, J. Comput. Phys. 336 (C) (2017) 36–50, doi:10.1016/j.jcp.2017.01.047.
[16] A. Damianou, N. Lawrence, Deep Gaussian processes, in: Artificial Intelligence and Statistics, 2013, pp. 207–215.
[17] A. Narayan, C. Gittelson, D. Xiu, A stochastic collocation algorithm with multifidelity models, SIAM J. Sci. Comput. 36 (2) (2014) A495–A521.
[18] W. Xing, M. Razi, R. Kirby, K. Sun, A. Shah, Greedy nonlinear autoregression for multifidelity computer models at different scales, Energy and AI 1 (2020) 100012.
[19] A. Lederer, J. Umlauft, S. Hirche, Uniform error bounds for Gaussian process regression with application to safe control, in: Proceedings of the Conference on Neural Information Processing Systems (NeurIPS), 2019.
[20] K. Cutajar, M. Pullin, A. Damianou, N. Lawrence, J. González, Deep Gaussian processes for multi-fidelity modeling, arXiv preprint arXiv:1903.07320(2019).
[21] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, MIT Press, 2006.
[22] M.C. Kennedy, A. O'Hagan, Bayesian calibration of computer models, J. R. Stat. Soc. Ser. B (Stat. Methodol.) 63 (3) (2001) 425–464.
[23] W. Xing, S.Y. Elhabian, V. Keshavarzzadeh, R.M. Kirby, Shared-GP: learning interpretable shared hidden structure across data spaces for design space analysis and exploration, J. Mech. Des. (2020) 1–16.
[24] M.A. Alvarez, L. Rosasco, N.D. Lawrence, et al., Kernels for vector-valued functions: a review, Found. Trends Mach. Learn. 4 (3) (2012) 195–266.
[25] W. Xing, F. Yu, P. Leung, X. Li, P. Wang, A. Shah, A new multi-task learning framework for fuel cell model outputs in high-dimensional spaces, J. Power Sources 482 (2021) 228930.
[26] J. Song, Y. Chen, Y. Yue, A general framework for multi-fidelity Bayesian optimization with Gaussian processes, in: Proceedings of the AISTATS, 2019.
[27] H. Salimbeni, M. Deisenroth, Doubly stochastic variational inference for deep gaussian processes, in: Proceedings of the Advances in Neural Information Processing Systems, 2017, pp. 4588–4599.
[28] M. Kanagawa, et al., Gaussian processes and kernel methods: a review on connections and equivalences., 2018 arXiv preprint arXiv:1807.02582.
[29] R. Schaback, Improved error bounds for scattered data interpolation by radial basis functions, Math. Comput. 68 (225) (1999) 201216.
[30] S. Mendelson, Improving the sample complexity using global data, IEEE Trans. Inf. Theor. 48 (7) (2006) 19771991.
[31] N. Srinivas, A. Krause, S.M. Kakade, M.W. Seeger, Information-theoretic regret bounds for Gaussian process optimization in the bandit setting, IEEE Trans. Inf. Theory 58 (5) (2012) 3250–3265.
[32] S.R. Chowdhury, A. Gopalan, On kernelized multi-armed bandits, in: D. Precup, Y.W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, Proceedings of Machine Learning Research, 70, 2017, pp. 844–853.
[33] N. De Freitas, A.J. Smola, M. Zoghi, Exponential regret bounds for Gaussian process bandits with deterministic observations, in: Proceedings of the 29th International Conference on Machine Learning, in: ICML'12, 2012, p. 955962.
[34] W. Wang, R. Tuo, C.F.J. Wu, On prediction properties of kriging: uniform error bounds and robustness, J. Am. Stat. Assoc. 115 (530) (2020) 920–930.
[35] A. Lederer, J. Umlauft, S. Hirche, Uniform error and posterior variance bounds for Gaussian process regression with application to safe control, 2021 arXiv preprint arXiv:2101.05328.
[36] R.J. Adler, J.E. Taylor, Random Fields and Geometry, Springer Science & Business Media, 2009.
[37] G. Santin, B. Haasdonk, Convergence rate of the data-independent p-greedy algorithm in kernel-based approximation, Dolomit. Res. Notes Approx. 10 (Special Issue) (2017).
[38] A. Paleyes, M. Pullin, M. Mahsereci, N. Lawrence, J. González, Emulation of physical processes with Emukit, in: Proceedings of the Second Workshop on Machine Learning and the Physical Sciences, NeurIPS, 2019.
[39] J.G. Lee, Computational Materials Science: An Introduction, CRC press, 2016.