Engineering Applications of Artificial Intelligence 67 (2018) 211–225

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Cope with diverse data structures in multi-fidelity modeling: A Gaussian process method

Haitao Liu [a,*], Yew-Soon Ong [b,c], Jianfei Cai [b], Yi Wang [d]

[a] Rolls-Royce@NTU Corporate Lab, Nanyang Technological University, Singapore 637460, Singapore
[b] School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore
[c] Data Science and Artificial Intelligence Research Center, Nanyang Technological University, Singapore 639798, Singapore
[d] Applied Technology Group, Rolls-Royce Singapore, 6 Seletar Aerospace Rise, Singapore 797565, Singapore

ARTICLE INFO

Keywords:
Multi-fidelity modeling
Gaussian process regression
diverse data structures
knowledge transfer

ABSTRACT

Multi-fidelity modeling (MFM) frameworks, especially the Bayesian MFM, have gained popularity in simulation based modeling, uncertainty quantification and optimization, due to the potential for reducing computational budget. In the view of multi-output modeling, the MFM approximates the high-/low-fidelity outputs simultaneously by considering the output correlations, and particularly, it transfers knowledge from the inexpensive low-fidelity outputs that have many training points to enhance the modeling of the expensive high-fidelity output that has a few training points. This article presents a novel multi-fidelity Gaussian process for modeling with diverse data structures. The diverse data structures mainly refer to the diversity of high-fidelity sample distributions, i.e., the high-fidelity points may randomly fill the domain, or more challengingly, they may cluster in some subregions. The proposed multi-fidelity model is composed of a global trend term and a local residual term. Particularly, the flexible residual term extracts both the shared and output-specific residual information via a data-driven weight parameter. Numerical experiments on two synthetic examples, an aircraft example and a stochastic incompressible flow example reveal that this very promising Bayesian MFM approach is capable of effectively extracting the low-fidelity information for facilitating the modeling of the high-fidelity output using diverse data structures.

1. Introduction

In recent years, new advances in computers and computing science lead to the widespread use of computer simulation models, e.g., computational fluid dynamics (CFD) and finite element analysis (FEA), in engineering design and optimization. In simulation based engineering problems, surrogates are starting to play an important role, since they can approximate the expensive simulation model at some training points for alleviating computational burden. Gaussian process regression (Rasmussen and Williams, 2006), also known as Kriging (Lophaven et al., 2002), is a widely used surrogate model, since it can provide not only the prediction response but also the related prediction variance.

This article focuses on a multi-fidelity scenario where the simulator for the physics-based problem of interests can be run at multiple levels of fidelity. The high fidelity (HF) simulator yields the most accurate predictions but is most time-consuming; whereas the fast low fidelity (LF) simulators provide coarse predictions, which however include the main features of the problem and thus are useful for preliminary exploration. The LF simulators are usually simplified analysis models by using coarse finite element meshes, relaxed boundary or convergence conditions, etc. For example, it was reported by Benamara et al. (2016) that the HF simulation for a 1.5 stage booster has 5 million meshes and requires 2 h; but the LF simulation has only 0.7 million meshes and requires only 15 min. In practice, we cannot afford extensive HF simulations at many training points but many LF simulations are affordable. Suppose that the simulator has Q levels of fidelity, the multi-fidelity modeling (MFM), also known as variable-fidelity modeling or data fusion, attempts to utilize the knowledge from the correlated yet inexpensive $Q-1$ LF simulators to enhance the modeling of the expensive HF simulator.

Considering Q levels of fidelity as Q correlated outputs, the information fusion can be achieved in the multi-output modeling framework. The multi-output GP (MOGP), also known as multi-variate Kriging (Kleijnen and Mehdad, 2014), has been developed and investigated

* Corresponding author.
E-mail addresses: htliu@ntu.edu.sg (H. Liu), ASYSOng@ntu.edu.sg (Y.-S. Ong), ASJFCai@ntu.edu.sg (J. Cai), Yi.Wang4@Rolls-Royce.com (Y. Wang).

https://doi.org/10.1016/j.engappai.2017.10.008
Received 30 May 2017; Received in revised form 28 August 2017; Accepted 5 October 2017
Available online 5 November 2017
0952-1976/

with a long history. The MOGP attempts to model multiple correlated outputs simultaneously by sharing the information across them, with the aim of outperforming individual modeling. The key in MOGP is to construct a valid multi-output covariance function to transfer useful information across outputs. A pioneer and well-known model developed in the field of geostatistics is called linear model of coregionalization (LMC) (Journel and Huijbregts, 1978). This model constructs valid covariance functions by a linear combination of several Gaussian processes. Thereafter, various MOGPs have been developed and extended in the context of LMC (Seeger et al., 2005; Bonilla et al., 2007; Hayashi et al., 2012; Osborne et al., 2012; Rakitsch et al., 2013; Dürichen et al., 2015; Hori et al., 2016). Another way to construct valid covariance functions is through process convolutions that convolve a base process, e.g., white Gaussian noise, with a smoothing kernel (Ver Hoef and Barry, 1998; Boyle and Frean, 2004; Álvarez and Lawrence, 2009, 2011).[1] The process convolutions can be regarded as a dynamic version of LMC (Álvarez and Lawrence, 2011; Álvarez et al., 2012).

In the multi-fidelity scenario, particularly, we attempt to use the inexpensive LF outputs to assist the modeling of the expensive HF output. Hence, compared to the typical MOGPs that share the information across the outputs, Kennedy and O'Hagan (2000) presented a Bayesian discrepancy-based MFM framework, which is an extension to the Co-Kriging model (Myers, 1982). In this framework, an auto-regressive model is proposed by expressing the HF output as the sum of the scaled LF outputs and an additive term that accounts for the discrepancy between the outputs, leading to not only the information sharing across outputs but also the asymmetric knowledge transfer from the LF outputs to the HF output. Later, Qian and Wu (2008) and Leen et al. (2012) provided an equivalent MFM framework in different views. Being a good multi-fidelity predictive model, Co-Kriging has been extended and improved, e.g., by reducing computational complexity (Le Gratiet and Garnier, 2014; Le Gratiet and Cannamela, 2015), using space-dependent scaling factor (Perdikaris et al., 2017), and incorporating gradient information (Han et al., 2013; Ulaganathan et al., 2015). Due to the remarkable performance, Co-Kriging has gained popularity in various fields, e.g., model inversion (Perdikaris and Karniadakis, 2016), uncertainty quantification (Perdikaris et al., 2015; Kennedy and O'Hagan, 2001), and multidisciplinary/robust/multi-objective optimization (Forrester et al., 2007; Keane, 2012; Han and Görtz, 2012; Kontogiannis et al., 2017). For more information about Co-Kriging and MFM, one can refer to the recent reviews and comparison studies (Fernández-Godino et al., 2016; Park et al., 2017; Toal, 2015).

In the context of Co-Kriging, it is usually assumed that we can control the sampling process such that the HF and LF training points spread over the entire domain evenly by for example the nested sampling strategy (Qian, 2009) and the nearest neighbor sampling strategy (Le Gratiet and Garnier, 2014). The space-filling nested data structure, though being beneficial for Co-Kriging, cannot always be available in practice. In realistic scenarios, we need to handle diverse data structures. The diverse data structures here mainly refer to the diversity of HF sample distributions, while the inexpensive LF outputs are assumed to have sufficient training points that cover the entire domain. For example, as shown in Fig. 1,[2] we have a set of uniformly distributed HF points, a set of randomly distributed HF points, and more challengingly, a set of partially distributed HF points clustered in a subregion. Besides, a practical example is that when using CFD solvers of different fidelities to simulate the flow around an aircraft, the inexpensive LF Euler simulation can be computed over the domain; while for saving computing time, the expensive HF Navier–Stokes simulation is only performed in flow regions with strong viscous effects. Hence, the diverse data structures, which contain different HF sample distributions, pose the demands of developing a particular multi-fidelity modeling approach that can effectively extract LF information to facilitate the HF modeling in different scenarios.

Therefore, this article presents a novel multi-fidelity GP model that is composed of a global trend term and a local residual term in order to tackle diverse data structures, e.g., full points that are available in the entire domain, or partial points that fill only some subregions. Particularly, in order to extract useful LF information effectively for facilitating the HF modeling, the local residual term contains a shared part and an output-specific part, the trade-off between which is dynamically determined by a data-driven weight parameter. The flexible model structure enables the approach to accomplish the multi-fidelity modeling well with diverse data structures.

The remainder of the article is organized as follows. Section 2 briefly introduces the single-output Gaussian process. Then, Section 3 presents the newly developed multi-fidelity Gaussian process in the MOGP framework. Thereafter, Section 4 comprehensively tests the new approach on two synthetic examples and two engineering examples with diverse characteristics and data structures. Finally, Section 5 offers some concluding remarks.

2. Single-output Gaussian process

Here we give a brief introduction to the single-output Gaussian process (SOGP). GP is a stochastic process wherein any finite subset of random variables follows a joint Gaussian distribution. As a non-parametric[3] model, the GP interprets the target function $f(\mathbf{x})$ where $\mathbf{x} \in R^d$ as a probability distribution in function space as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \tag{1}$$

which is completely defined by the mean function $m(\mathbf{x})$ that is usually taken as zero without loss of generality, and the covariance function $k(\mathbf{x}, \mathbf{x}')$. In practice, we usually use the squared exponential (SE) covariance function as

$$k_{SE}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{x}')^T P^{-1}(\mathbf{x}-\mathbf{x}')\right), \tag{2}$$

where the signal variance σ_f^2 represents an output scale amplitude; the ith element of the diagonal matrix $P \in R^{d\times d}$ is the characteristic length-scale l_i^2 that controls the width of the bell-shaped curve along the ith dimension. For other well-known covariance functions, e.g., the Matérn covariance function and the rational quadratic covariance function, one can refer to Rasmussen and Williams (2006).

Typically, in many realistic scenarios, instead of the latent function values themselves, we only have the observed response

$$y(\mathbf{x}) = f(\mathbf{x}) + \epsilon, \tag{3}$$

where the independent and identically distributed (*i.i.d.*) noise $\epsilon \sim \mathcal{N}(0, \sigma_s^2)$ accounts for the practical measurement errors, the modeling errors, the manufacturing tolerances, etc. It has been pointed out that we can gain benefits from the consideration of noise in GP for numerical stability (Ababou et al., 1994; Neal, 1997) and better statistical properties, e.g., prediction accuracy and coverage (Gramacy and Lee, 2012).

For the target function $f(\mathbf{x})$, suppose that we have a set of training points $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}^T$ in the design space $D \in [0,1]^d$, and their corresponding output observations $\mathbf{y} = \{y(\mathbf{x}_1), \ldots, y(\mathbf{x}_n)\}^T$. The joint prior distribution of the observed dataset $D = \{X, \mathbf{y}\}$ augmented with a test data $\{\mathbf{x}_*, f_*\}$ is as

$$\begin{bmatrix}\mathbf{y}\\ f_*\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\mathbf{0}\\ 0\end{bmatrix}, \begin{bmatrix}K(X,X)+\sigma_s^2 I & K(X,\mathbf{x}_*)\\ K(\mathbf{x}_*,X) & k(\mathbf{x}_*,\mathbf{x}_*)\end{bmatrix}\right), \tag{4}$$

[1] If the base process is a Gaussian process, then the convolved process is ensured to be a Gaussian process.

[2] The HF and LF functions in this 1D multi-fidelity example are expressed by Eqs. (45) and (46), respectively.

[3] "Non-parametric" means that the GP has no explicit parameters to control the functional form of the model. But it still has some *hyperparameters* that need to be inferred in the modeling process.

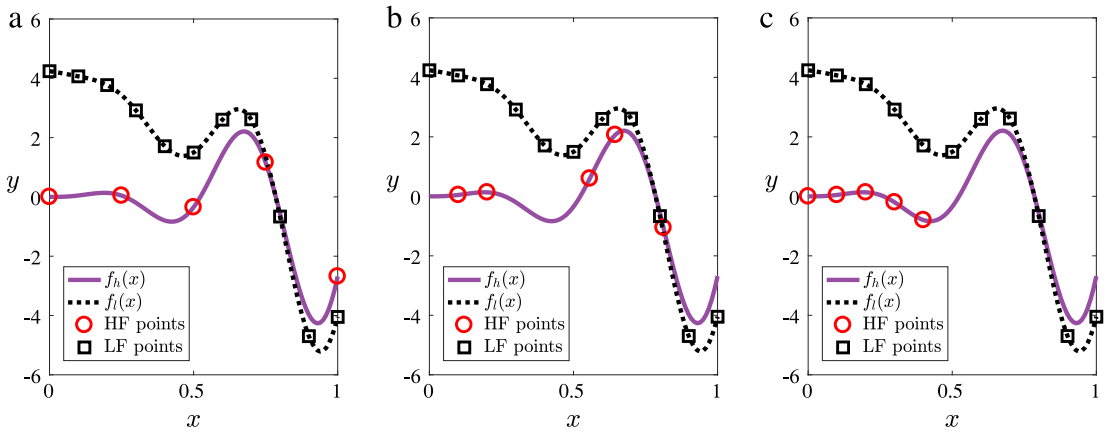

Fig. 1. A 1D multi-fidelity example with diverse data structures: (a) a uniform HF sample distribution, (b) a random HF sample distribution and (c) a partial HF sample distribution.

where the symmetric and positive semi-definite (PSD) covariance matrix $K(X,X)$ is

$$K(X,X)=\begin{bmatrix} k(\mathbf{x}_1,\mathbf{x}_1) & \cdots & k(\mathbf{x}_1,\mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n,\mathbf{x}_1) & \cdots & k(\mathbf{x}_n,\mathbf{x}_n) \end{bmatrix}. \tag{5}$$

By conditioning the joint Gaussian prior distribution on the observations, we obtain the posterior distribution of f_* as

$$f_*|X,\mathbf{y},\mathbf{x}_* \sim \mathcal{N}(\hat{f}_*,\sigma_*^2). \tag{6}$$

The prediction mean $\hat{f}_*$ and the prediction variance σ_*^2 are respectively given as

$$\hat{f}_* = \mathbf{k}_*^T[K(X,X)+\sigma_s^2 I]^{-1}\mathbf{y}, \tag{7}$$

$$\sigma_*^2 = k(\mathbf{x}_*,\mathbf{x}_*)-\mathbf{k}_*^T[K(X,X)+\sigma_s^2 I]^{-1}\mathbf{k}_*, \tag{8}$$

where the $n\times 1$ vector $\mathbf{k}_* = K(X,\mathbf{x}_*)$ denotes the covariance between the test point $\mathbf{x}_*$ and the n training points. Note that for obtaining the prediction variance of y_*, we can simply add σ_s^2 to Eq. (8).

In order to use Eqs. (7) and (8) for prediction, we need to obtain the hyperparameters $\boldsymbol{\theta}=\{\sigma_f^2,l_1,\ldots,l_d,\sigma_s^2\}^T$, which can be inferred by minimizing the negative log marginal likelihood (NLML) as

$$\boldsymbol{\theta}_{\text{opt}} = \arg\min_{\boldsymbol{\theta}} \text{NLML}, \tag{9}$$

where

$$\begin{aligned}\text{NLML} &= -\log p(\mathbf{y}|X,\boldsymbol{\theta}) \\ &= \frac{1}{2}\mathbf{y}^T[K(X,X)+\sigma_s^2 I]^{-1}\mathbf{y} \\ &\quad + \frac{1}{2}\log|K(X,X)+\sigma_s^2 I| + \frac{n}{2}\log 2\pi.\end{aligned} \tag{10}$$

It is found that the NLML expression automatically achieves a bias–variance tradeoff: the first data-fit term on the right-hand side of Eq. (10) penalizes low data likelihood; the second term penalizes model complexity; and the last term is a normalization constant (Rasmussen and Williams, 2006). By pre-calculating the partial derivatives of the marginal likelihood w.r.t. the hyperparameters $\boldsymbol{\theta}$, we can use the efficient gradient descent algorithm to solve problem (9) for inferring the $\boldsymbol{\theta}_{\text{opt}}$.

3. Multi-fidelity Gaussian process

3.1. General multi-output Gaussian process

Different from the single-output case, here we consider the multi-output case that is frequently encountered in practice. In the multi-output scenario, we attempt to learn the mapping between the input space $D\in R^d$ and the output space R^Q where $Q>1$ is the number of outputs. The multi-output GP (MOGP) is to model Q outputs $\{f_t\}_{t=1}^Q$ simultaneously by considering the output correlations, with the aim of outperforming individual modeling.

The construction of MOGP is similar to that of SOGP. As stated before, a Gaussian process is defined as a random field wherein any finite number of random variables follow a joint Gaussian distribution. For the single-output case, the random variables evaluated at different points are associated to a single process f. For the multi-output case, they are associated to Q different processes $\{f_t\}_{t=1}^Q$. Hence, the Q outputs $\mathbf{f}=\{f_1,\ldots,f_Q\}^T$ are assumed to follow a Gaussian process as

$$\mathbf{f}(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0},\mathcal{K}_M(\mathbf{x},\mathbf{x}')), \tag{11}$$

where the multi-output covariance $\mathcal{K}_M(\mathbf{x},\mathbf{x}')$ is defined as

$$\mathcal{K}_M(\mathbf{x},\mathbf{x}')=\begin{bmatrix} k_{11}(\mathbf{x},\mathbf{x}') & \cdots & k_{1Q}(\mathbf{x},\mathbf{x}') \\ \vdots & \ddots & \vdots \\ k_{Q1}(\mathbf{x},\mathbf{x}') & \cdots & k_{QQ}(\mathbf{x},\mathbf{x}') \end{bmatrix}. \tag{12}$$

The entry $k_{t,t'}(\mathbf{x},\mathbf{x}')$ corresponds to the covariance between outputs $f_t(\mathbf{x})$ and $f_{t'}(\mathbf{x}')$. It represents the degree of correlation or similarity between the two outputs.

In the multi-output scenario, we assume that $X=\{\{\mathbf{x}_{t,i}\}_{i=1}^{n_t}\}_{t=1}^Q$ and $\mathbf{y}=\{\{y_{t,i}\}_{i=1}^{n_t}\}_{t=1}^Q$ are the collection of training points and observations for the Q outputs, and we have $N=\sum_{t=1}^Q n_t$. Then, the matrix $X\in R^{N\times d}$ has Q blocks with $X_t=\{\mathbf{x}_{t,1},\ldots,\mathbf{x}_{t,n_t}\}^T$ corresponding to the training set for output f_t; the vector $\mathbf{y}\in R^{N\times 1}$ also has Q components with $\mathbf{y}_t=\{y_{t,1},\ldots,y_{t,n_t}\}^T$ corresponding to the observations of f_t at X_t. Since the point $\mathbf{x}_{t,i}$ and the observation $y_{t,i}$ are related to output f_t, we employ an output indicator $c_t=t$ as an additional input to the model. For the convenience of presentation below, we assume that $X_1=\cdots=X_Q=\overline{X}\in R^{n\times d}$, i.e., $n_t=n$ and $\mathbf{x}_{t,i}=\mathbf{x}_i$ for $t=1,\ldots,Q$.[4]

Similarly, for output f_t, we consider a regression model

$$y_t(\mathbf{x}) = f_t(\mathbf{x})+\epsilon_t, \tag{13}$$

where the *i.i.d.* noise $\epsilon_t\sim\mathcal{N}(0,\sigma_{s,t}^2)$. Thereafter, the likelihood function for the Q outputs follows

$$p(\mathbf{y}|\mathbf{f},\mathbf{x},\Sigma_s)=\mathcal{N}(\mathbf{f}(\mathbf{x}),\Sigma_s), \tag{14}$$

where $\Sigma_s\in R^{Q\times Q}$ is a diagonal matrix with elements $\{\sigma_{s,t}^2\}_{t=1}^Q$. Then, the posterior distribution for new observations $\mathbf{f}_*=\{f_1(\mathbf{x}_*),\ldots,f_Q(\mathbf{x}_*)\}^T$ given the training data $\{X,\mathbf{y}\}$ and a test point $\mathbf{x}_*$ can be analytically expressed as

$$\mathbf{f}_*|X,\mathbf{y},\mathbf{x}_* \sim \mathcal{N}(\hat{\mathbf{f}}_*,\Sigma_*). \tag{15}$$

[4] The MOGP can be readily extended to the general situation where $X_1\neq\cdots\neq X_Q$.

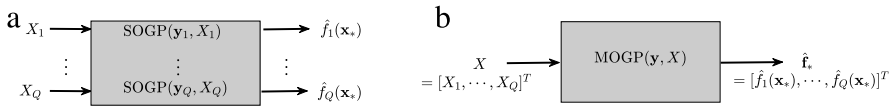

Fig. 2. Illustration of (a) multiple SOGP modeling process and (b) MOGP modeling process.

The prediction mean and variance are respectively given as

$$\hat{\mathbf{f}}_* = K_{M*}^T [K_M(\overline{X}, \overline{X}) + \Sigma_M]^{-1} \mathbf{y}, \tag{16}$$

$$\Sigma_* = \mathcal{K}_M(\mathbf{x}_*, \mathbf{x}_*) - K_{M*}^T [K_M(\overline{X}, \overline{X}) + \Sigma_M]^{-1} K_{M*}, \tag{17}$$

where $K_M(\overline{X}, \overline{X}) \in R^{N \times N}$ is expressed as

$$K_M(\overline{X}, \overline{X}) = \begin{bmatrix} K_{11}(\overline{X}, \overline{X}) & \cdots & K_{1Q}(\overline{X}, \overline{X}) \\ \vdots & \ddots & \vdots \\ K_{Q1}(\overline{X}, \overline{X}) & \cdots & K_{QQ}(\overline{X}, \overline{X}) \end{bmatrix}. \tag{18}$$

It is a symmetric and block partitioned matrix calculated by Eq. (12); $K_{M*} = K_M(\overline{X}, \mathbf{x}_*) \in R^{N \times Q}$ has entries $k_{t,t'}(\mathbf{x}_i, \mathbf{x}_*)$ for $i = 1, \ldots, n$ and $t, t' = 1, \ldots, Q$; $\mathcal{K}_M(\mathbf{x}_*, \mathbf{x}_*) \in R^{Q \times Q}$ has elements $k_{t,t'}(\mathbf{x}_*, \mathbf{x}_*)$ for $t, t' = 1, \ldots, Q$; $\Sigma_M = \Sigma_s \otimes I_n \in R^{N \times N}$ is a diagonal noise matrix; and the tth diagonal element of Σ_* corresponds to $\sigma_t^2(\mathbf{x}_*)$.

Fig. 2 depicts the general modeling processes of SOGP and MOGP for a multi-output regression problem, respectively. It is found that compared to SOGP, MOGP is promising to improve the prediction quality by considering the outputs simultaneously. A main drawback of MOGP is the increased computational complexity. The SOGP only needs to calculate the inverse of an $n \times n$ covariance matrix in Eq. (5); the MOGP, however, has to calculate the inverse of an $N \times N$ covariance matrix in Eq. (18). Besides, by considering the Q outputs jointly, the number of hyperparameters in MOGP increases rapidly, which makes the min-NLML problem (9) a non-trivial high-dimensional optimization task.

It is found that the key in MOGP is to construct admissible covariance structure $k_{tt'}(\mathbf{x}, \mathbf{x}')$ in order to (1) make the covariance matrix K_M positive definite, and (2) capture the output correlations. Next, we focus on the multi-fidelity problem with outputs yielding different levels of fidelity. We particularly propose a multi-fidelity GP (MFGP) model that can effectively transfer knowledge from the inexpensive LF outputs to the expensive HF output in cases with diverse data structures.

3.2. Proposed MFGP model

For the convenience of presentation, we here consider the case with two-level fidelity ($Q = 2$). In this context, (1) the HF output f_h provides accurate predictions but requires huge computational budget. It means we cannot afford to generate a considerable number of HF points for building a reliable HF model; (2) the LF output f_l is cheap to run but provides coarse predictions. It means we can have enough LF points to build a well-fitted LF model; and (3) the HF output and the LF output are correlated, since the LF output can capture the main features of the studied problem.

Conceptually, we decompose an output as

$$f(\mathbf{x}) = f_T(\mathbf{x}) + f_R(\mathbf{x}), \tag{19}$$

where $f_T(\mathbf{x})$, denoted as *global trend*, captures the global features of $f(\mathbf{x})$; while $f_R(\mathbf{x})$, denoted as *local residual*, captures the local residual features of $f(\mathbf{x})$. For the correlated f_h and f_l, they are respectively expressed in a linear combination form by using the *trend-residual* decomposition in Eq. (19) as

$$f_h(\mathbf{x}) = a_h u_T(\mathbf{x}) + u_{R,h}(\mathbf{x}), \tag{20}$$

$$f_l(\mathbf{x}) = a_l u_T(\mathbf{x}) + u_{R,l}(\mathbf{x}), \tag{21}$$

where the latent function $u_T(\mathbf{x})$ that is same for the two outputs represents the similar global features (common features) of f_h and f_l; a_h and a_l are the HF and LF correlation parameters for $u_T(\mathbf{x})$, respectively; and the latent functions $u_{R,h}(\mathbf{x})$ and $u_{R,l}(\mathbf{x})$ that are different for the two outputs capture the HF and LF local features, respectively. Each of these latent functions can be interpreted as a Gaussian distribution in the function space as

$$u_T(\mathbf{x}) \sim \mathcal{GP}(0, k_T(\mathbf{x}, \mathbf{x}')), \tag{22}$$

$$u_{R,h}(\mathbf{x}) \sim \mathcal{GP}(0, k_{R,h}(\mathbf{x}, \mathbf{x}')), \tag{23}$$

$$u_{R,l}(\mathbf{x}) \sim \mathcal{GP}(0, k_{R,l}(\mathbf{x}, \mathbf{x}')). \tag{24}$$

Similar to the typical MOGPs (Kennedy and O'Hagan, 2000; Seeger et al., 2005; Osborne et al., 2012; Álvarez et al., 2012), we assume that the three Gaussian processes are independent from each other, i.e., $\mathrm{cov}[u_T(\mathbf{x}), u_{R,h}(\mathbf{x}')] = \mathrm{cov}[u_T(\mathbf{x}), u_{R,l}(\mathbf{x}')] = \mathrm{cov}[u_{R,h}(\mathbf{x}), u_{R,l}(\mathbf{x}')] = 0$. The independent assumption allows us to derive a concise covariance function and reduce the model complexity.[5]

It can be seen that the latent process u_T shares the same hyperparameters for f_h and f_l. This enables the model to capture the common features of f_h and f_l; but it is a strong constraint, leaving no separate parameters to describe the correlations (i.e., the variability) of the HF and LF common features. Hence, the correlation parameters a_h and a_l are used to enhance the flexibility in learning the common features via scaling the HF and LF outputs. Furthermore, f_h and f_l adopt different residual processes $u_{R,h}$ and $u_{R,l}$ to account for their output-specific features. These residual processes have individual hyperparameters, thus bringing great flexibility in modeling the output-specific features. Besides, these residual processes can help reduce negative transfer across outputs (Leen et al., 2012).

As mentioned before, we assume that the GP model for f_l can be well fitted due to the relatively cheap LF simulation. On the contrary, the basic assumption for f_h is that we only have little HF information since the HF simulation is time-consuming. The lack of HF information is caused by two facts: (1) the number of HF points is limited; and (2) the distribution of HF points is undesirable. The first fact is an inherent property in multi-fidelity problems and is also the motivation for multi-fidelity modeling. While the second fact is rarely considered since it is usually assumed that the sampling process is controllable so that the HF training points are ensured to fill the entire domain evenly. The space-filling points indeed help extract the LF information effectively for good multi-fidelity modeling (Forrester et al., 2007; Han and Görtz, 2012; Le Gratiet and Garnier, 2014; Perdikaris et al., 2017). However, in practice we typically do not have the access to the generation of training points, but merely receive the HF and LF data that may have diverse data structures from the customers. For example, as shown in Fig. 1, the provided HF points may distribute randomly in the design space. More challengingly, the HF points may cluster together in a subregion. In this case, we have to tackle an intractable extrapolation problem. Therefore, the diverse data structures result in further HF information loss and harder multi-fidelity modeling.

For the model in Eqs. (20) and (21), it is found that the quality of the residual process $u_{R,h}(\mathbf{x})$ is greatly affected by the distribution of HF points. For the space-filling distribution in Fig. 1(a), it is possible to build a valid HF residual model via a purely output-specific process $u_{R,h}(\mathbf{x})$. However, for the partial HF sample distribution in Fig. 1(c), due to the missing of HF points in the range [0.5, 1.0], it is hard to learn a valid HF residual model by the completely free $u_{R,h}(\mathbf{x})$. In this scenario, $u_{R,h}(\mathbf{x})$ is modeled by using only the HF points in [0.0, 0.5], and then it has to

[5] The relaxation of the independent assumption has been explored in some literature (Vargas-Guzmán et al., 2002).

extrapolate the predictions in [0.5, 1.0], which, however, is a non-trivial task.

To address this issue, an alternative way is to seek the help from the correlated LF output. Along this line, the simplest way is to transfer the residual information of f_l as $u_{R,h}(\mathbf{x}) = u_{R,l}(\mathbf{x}) = u_R(\mathbf{x})$. Since now the HF and LF residual processes are fixed to be the same, the HF output will completely learn the LF residual features.

But this kind of relatedness is too strong for the HF residual process, since it is confined to be the same as the LF residual process. Therefore, to allow for the flexibility in capturing the possible shared information among the residual processes as well as the output-specific features, we relax $u_{R,h}(\mathbf{x})$ and $u_{R,l}(\mathbf{x})$ as

$$u_{R,h}(\mathbf{x}) = \rho u_R(\mathbf{x}) + (1-\rho)u^s_{R,h}(\mathbf{x}), \tag{25}$$

$$u_{R,l}(\mathbf{x}) = \rho u_R(\mathbf{x}) + (1-\rho)u^s_{R,l}(\mathbf{x}), \tag{26}$$

where $u_R(\mathbf{x})$ corresponds to the information shared between the HF and LF residuals; and $u^s_{R,h}(\mathbf{x})$ and $u^s_{R,l}(\mathbf{x})$ model the output-specific features of f_h and f_l, respectively. Given the training set X, we should decide how much information can be shared between the HF and LF residual processes. To this end, we introduce a weight parameter ρ to represent the conflict between the shared residual process $u_R(\mathbf{x})$ and the output-specific residual processes $u^s_{R,h}(\mathbf{x})$ and $u^s_{R,l}(\mathbf{x})$. That is, a large ρ indicates that the shared residual process contributes more to the HF and LF residual processes. The role of ρ can be further identified in terms of residual information fusion in the multi-fidelity covariance (32), which will be discussed below.

Note that we use the same ρ for $u_{R,h}(\mathbf{x})$ and $u_{R,l}(\mathbf{x})$, because the residual processes have enough individual parameters in $u^s_{R,h}(\mathbf{x})$ and $u^s_{R,l}(\mathbf{x})$ to describe the output-specific features. Besides, it is worth noting that the ρ value is not limited to [0.0, 1.0]. Like the "free-form" parameterization introduced below in Eq. (39), we treat ρ as a hyperparameter and let it scale freely, and infer the optimal value by minimizing the NLML in Eq. (9).

Now the proposed MFGP model can be expressed as

$$f_h(\mathbf{x}) = a_h u_T(\mathbf{x}) + \rho u_R(\mathbf{x}) + (1-\rho)u^s_{R,h}(\mathbf{x}), \tag{27}$$

$$f_l(\mathbf{x}) = a_l u_T(\mathbf{x}) + \rho u_R(\mathbf{x}) + (1-\rho)u^s_{R,l}(\mathbf{x}). \tag{28}$$

Based on the independent assumption, we derive the self-covariance function for f_h as

$$\begin{aligned}\mathrm{cov}[f_h(\mathbf{x}), f_h(\mathbf{x}')] &= k_{hh}(\mathbf{x},\mathbf{x}') \\ &= a_h^2 k_T(\mathbf{x},\mathbf{x}') + \rho^2 k_R(\mathbf{x},\mathbf{x}') + (1-\rho)^2 k^s_{R,h}(\mathbf{x},\mathbf{x}'),\end{aligned} \tag{29}$$

and the self-covariance function for f_l as

$$\begin{aligned}\mathrm{cov}[f_l(\mathbf{x}), f_l(\mathbf{x}')] &= k_{ll}(\mathbf{x},\mathbf{x}') \\ &= a_l^2 k_T(\mathbf{x},\mathbf{x}') + \rho^2 k_R(\mathbf{x},\mathbf{x}') + (1-\rho)^2 k^s_{R,l}(\mathbf{x},\mathbf{x}'),\end{aligned} \tag{30}$$

and finally the cross-covariance function as

$$\begin{aligned}\mathrm{cov}[f_h(\mathbf{x}), f_l(\mathbf{x}')] &= k_{hl}(\mathbf{x},\mathbf{x}') = k_{lh}(\mathbf{x},\mathbf{x}') \\ &= a_h a_l k_T(\mathbf{x},\mathbf{x}') + \rho^2 k_R(\mathbf{x},\mathbf{x}').\end{aligned} \tag{31}$$

Consequently, the multi-fidelity covariance $\mathcal{K}_M(\mathbf{x},\mathbf{x}')$ can be formulated as

$$\begin{aligned}\mathcal{K}_M(\mathbf{x},\mathbf{x}') &= \begin{bmatrix} k_{hh}(\mathbf{x},\mathbf{x}') & k_{hl}(\mathbf{x},\mathbf{x}') \\ k_{lh}(\mathbf{x},\mathbf{x}') & k_{ll}(\mathbf{x},\mathbf{x}')\end{bmatrix} \\ &= \begin{bmatrix} a_h^2 k_T(\mathbf{x},\mathbf{x}') & a_h a_l k_T(\mathbf{x},\mathbf{x}') \\ a_h a_l k_T(\mathbf{x},\mathbf{x}') & a_l^2 k_T(\mathbf{x},\mathbf{x}')\end{bmatrix} + \rho^2 \begin{bmatrix} k_R(\mathbf{x},\mathbf{x}') & k_R(\mathbf{x},\mathbf{x}') \\ k_R(\mathbf{x},\mathbf{x}') & k_R(\mathbf{x},\mathbf{x}')\end{bmatrix} \\ &\quad + (1-\rho)^2 \begin{bmatrix} k^s_{R,h}(\mathbf{x},\mathbf{x}') & 0 \\ 0 & k^s_{R,l}(\mathbf{x},\mathbf{x}')\end{bmatrix}.\end{aligned} \tag{32}$$

The residual parts in Eq. (32) can be written as $\rho^2\mathcal{K}_R(\mathbf{x},\mathbf{x}') + (1-\rho^2)\mathcal{K}^s_R(\mathbf{x},\mathbf{x}')$, from which we can clearly see the role of parameter ρ for the residual information fusion. It is observed that the MFGP assigns a weight of ρ^2 to the shared residual part $\mathcal{K}_R$, and a weight of $(1-\rho)^2$ to the output-specific residual part $\mathcal{K}^s_R$. Therefore, to quantify the contribution of the shared residual process $u_R(\mathbf{x})$ to the total residual process $\rho u_R(\mathbf{x}) + (1-\rho)u^s_{R,t}(\mathbf{x})$ in the MFGP model, we introduce a normalized parameter as

$$\overline{\rho^2} = \frac{\rho^2}{\rho^2 + (1-\rho)^2} \in [0,1]. \tag{33}$$

The extreme value $\overline{\rho^2} = 1$ indicates that the covariance $\mathcal{K}_M$ completely uses the information shared from the LF residual to represent the HF residual; on the contrary, $\overline{\rho^2} = 0$ indicates that $\mathcal{K}_M$ completely uses the output-specific features to represent the HF residual; the tradeoff value $0 < \overline{\rho^2} < 1$ indicates the simultaneous utilization of shared and output-specific residual information. In practice, for example, if the HF points distribute like Fig. 1(a), we prefer using more information from $u^s_{R,h}(\mathbf{x})$ and $u^s_{R,l}(\mathbf{x})$ to capture their specific features by adopting a low $\overline{\rho^2}$ value; on the contrary, if the HF points distribute like Fig. 1(b) and even Fig. 1(c), we need to transfer some residual knowledge from f_l to improve the HF predictions in regions with no HF points by adopting a high $\overline{\rho^2}$ value. Therefore, the data-driven parameter $\overline{\rho^2}$, which represents the mode of residual information utilization for MFGP, depends on the training set, i.e., $\overline{\rho^2} \propto X$.

Finally, given the multi-fidelity covariance in Eq. (32) and the training data X, we can use Eq. (18) to obtain the covariance matrix, and then substitute it into Eqs. (16) and (17) to obtain the prediction mean and variance of each output at a test point $\mathbf{x}_*$.

3.3. Q -level MFGP model

Here, we offer a generalized version of MFGP for handling Q-level multi-fidelity problems. Suppose that the Q outputs $\{f_t\}_{t=1}^Q$ are sorted by increasing fidelity, i.e., f_Q has the highest fidelity and f_1 has the lowest fidelity. The generalization is straightforward by modeling each of the outputs as a linear combination of several Gaussian processes as

$$f_t(\mathbf{x}) = a_t u_T(\mathbf{x}) + \rho u_R(\mathbf{x}) + (1-\rho)u^s_{R,t}(\mathbf{x}), \quad 1 \le t \le Q. \tag{34}$$

The matrix formulation of the Q outputs can be expressed as

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_Q(\mathbf{x})\end{bmatrix} = [\mathbf{a} \;\; \rho \;\; \rho']\mathbf{u}, \tag{35}$$

where the $Q \times 1$ vector $\mathbf{a} = [a_1, \ldots, a_Q]^T$ contains the correlation parameters, the $Q \times 1$ vector $\rho = \rho\mathbf{1}$'s and the $Q \times Q$ diagonal matrix $\rho' = (1-\rho)I$ contain the weight parameters, and the $(Q+2) \times 1$ vector $\mathbf{u} = [u_T(\mathbf{x}), u_R(\mathbf{x}), u^s_{R,1}(\mathbf{x}), \ldots, u^s_{R,Q}(\mathbf{x})]^T$ contains the trend, shared residual and output-specific residual Gaussian processes.

The covariance between $f_t(\mathbf{x})$ and $f_t(\mathbf{x}')$ is expressed as

$$k_{tt}(\mathbf{x},\mathbf{x}') = a_t^2 k_T(\mathbf{x},\mathbf{x}') + \rho^2 k_R(\mathbf{x},\mathbf{x}') + (1-\rho)^2 k^s_{R,t}(\mathbf{x},\mathbf{x}'), \tag{36}$$

and the cross-covariance between $f_t(\mathbf{x})$ and $f_{t'}(\mathbf{x}')$ is

$$k_{tt'}(\mathbf{x},\mathbf{x}') = a_t a_{t'} k_T(\mathbf{x},\mathbf{x}') + \rho^2 k_R(\mathbf{x},\mathbf{x}'). \tag{37}$$

Then we can obtain the multi-fidelity covariance $\mathcal{K}_M(\mathbf{x},\mathbf{x}')$ with the (t,t') element as $k_{tt'}(\mathbf{x},\mathbf{x}')$ for $1 \le t,t' \le Q$. Finally, given the training set X, we can train this Q-level MFGP model and use it for predicting the Q outputs at a test point $\mathbf{x}_*$ simultaneously via Eqs. (16) and (17).

3.4. Learning hyperparameters

For the convenience of presentation, recall the assumption that $X_1 = \cdots = X_Q = \overline{X}$, and then similar to Eq. (32) we can decompose the covariance matrix as

$$\begin{aligned}K_M(\overline{X},\overline{X}) =& A \otimes K_T(\overline{X},\overline{X}) + \rho^2 B \otimes K_R(\overline{X},\overline{X}) \\ &+ (1-\rho)^2 \mathrm{diag}[K^s_{R,1}(\overline{X},\overline{X}), \ldots, K^s_{R,Q}(\overline{X},\overline{X})],\end{aligned} \tag{38}$$

where $K_T(\overline{X},\overline{X})$, $K_R(\overline{X},\overline{X})$, $K^s_{R,1}(\overline{X},\overline{X})$, $\cdots$, $K^s_{R,Q}(\overline{X},\overline{X})$ represent the $n \times n$ trend, shared residual and output-specific residual covariance

matrices, respectively; $A = \mathbf{a}\mathbf{a}^T$ is the output correlation matrix with the element $A_{tt'} = a_t a_{t'}$; and $B \in R^{Q\times Q}$ is a matrix with all the elements as one. It is found that the diagonal elements of A describe the self-correlation of Q outputs, while the non-diagonal elements describe the correlation between two different outputs. Specifically, if A is an identity matrix, the Q outputs would be treated independently in the global trend term of Eq. (38).

More generally, the correlation matrix A can be regarded as a covariance matrix wherein the element $a_{tt'} = k_a(c_t, c_{t'})$ depends on the output indicators c_t and $c_{t'}$. In this context, to ensure that the matrix A is PSD, we employ a "free-form" strategy (Bonilla et al., 2007) to parameterize it. Based on the Cholesky decomposition $A = LL^T$, we parameterize the lower triangular matrix L as

$$L = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ a_2 & a_3 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ a_{w-Q+1} & a_{w-Q+2} & \cdots & a_w \end{bmatrix}, \tag{39}$$

where $w = Q(Q+1)/2$ is the number of correlation parameters. One advantage of the free-form parameterization is that the elements of A are free to scale individually for each pair of outputs, which enhances the ability to well estimate the output correlations (Dürichen et al., 2015).

It is found that the number of correlation parameters in a full rank A is $Q(Q+1)/2$, which grows quadratically with the number of outputs and leads to many optimization parameters to estimate. Alternatively, Bonilla et al. (2007) suggested using a rank-P approximation of A based on an incomplete Cholesky decomposition as

$$A \approx \widetilde{A} = \widetilde{L}\widetilde{L}^T, \tag{40}$$

where $\widetilde{L}$ is a $Q \times P$ ($P \leq Q$) matrix. With the increase of P, the correlation matrix approximated by Eq. (40) is promising for capturing the output correlations more accurately and flexibly, but requiring more computational cost because of the increasing number of correlation parameters. It is found that the parameterization $\widetilde{A} = \mathbf{a}\mathbf{a}^T$ in Eq. (38) is actually a rank-1 approximation of A.

In the numerical experiments conducted in Section 4, we test the MFGP approach for problems with up to three-level fidelity. Hence, the full rank A is employed in order to well estimate the output correlations. Suppose that all the Gaussian processes $\{u_T, u_R, u^s_{R,1}, \ldots, u^s_{R,Q}\}$ choose the SE covariance function in Eq. (2), given the training data X we need to infer the hyperparameters θ including w correlation parameters $\{a_i\}_{i=1}^w$ in A, $d+1$ covariance parameters $\{\sigma_f^2, l_1, \ldots, l_d\}$ for each of the $Q+2$ covariance functions, Q noise variances $\{\sigma_{s,t}^2\}_{t=1}^Q$, and finally the weight parameter ρ. Similar to the SOGP, these hyperparameters can be inferred by maximizing the marginal likelihood $p(\mathbf{y}|X,\theta)$. The partial derivatives of the NLML w.r.t. the hyperparameters are derived as

$$\begin{aligned} \frac{\partial \mathrm{NLML}}{\partial \theta_j} &= -\frac{1}{2}\mathbf{y}^T K_y^{-1}\frac{\partial K_y}{\partial \theta_j}K_y^{-1}\mathbf{y} + \frac{1}{2}\mathrm{tr}\left(K_y^{-1}\frac{\partial K_y}{\partial \theta_j}\right) \\ &= \frac{1}{2}\mathrm{tr}\left((K_y^{-1} - \boldsymbol{\alpha}\boldsymbol{\alpha}^T)\frac{\partial K_y}{\partial \theta_j}\right), \end{aligned} \tag{41}$$

where $K_y = K_M(\overline{X},\overline{X}) + \Sigma_M$ and $\boldsymbol{\alpha} = [K_M(\overline{X},\overline{X}) + \Sigma_M]^{-1}\mathbf{y}$. Straightforwardly, with the derivative information, we employ the gradient descent algorithm to solve the auxiliary optimization problem (9) for inferring the hyperparameters efficiently.

3.5. Workflow of MFGP

Fig. 3 depicts the MFGP modeling process wherein we use a rank-1 correlation matrix $\widetilde{A} = \mathbf{a}\mathbf{a}^T$ for illustration. It is found that each output $y_{t,i}$ is expressed as a linear combination of four stationary Gaussian processes including the global trend process u_T, the shared residual process u_R, the output-specific residual process $u^s_{R,t}$, and the noise process ϵ_t. The two shared processes enable MFGP to transfer the common global and local features from the LF outputs to enhance the modeling of the HF output, whereas the output-specific processes account for the specific features of each output. The correlation parameters $\{a_t\}_{t=1}^Q$ and the weight parameter ρ bring flexibility in further improving the modeling of the Q outputs.

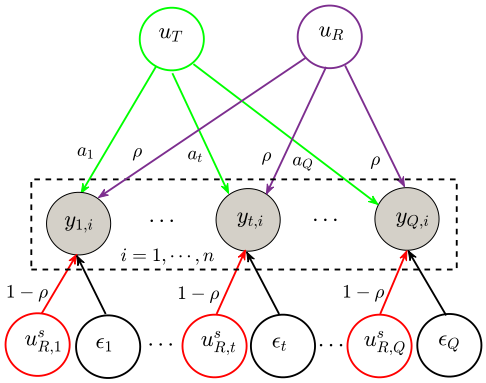

Fig. 3. Graphical model of the MFGP modeling process.

Given a training set X and the associated observations $\mathbf{y}$ from Q outputs sorted by increasing fidelity, the workflow of MFGP is elaborated below.

Step 1 Preparation. Before constructing the MFGP model, we normalize all the outputs with zero mean and unit variance. This normalization is necessary because the Q outputs may have different signal strengths (Bilionis and Zabaras, 2012). Besides, we augment the data point $\mathbf{x}_{t,i}$ as $\{\mathbf{x}_{t,i}, c_t\}$ for $t = 1, \ldots, Q$ and $i = 1, \ldots, n_t$, where the role of c_t in the modeling process is only to indicate the belonging of point $\mathbf{x}_{t,i}$.

Step 2 Model training. We choose the SE covariance function in Eq. (2) for the processes u_T, u_R and $\{u^s_{R,t}\}_{t=1}^Q$. Given the training set X, the full rank correlation matrix A parameterized by Eq. (39) and the weight parameter ρ, we construct the multi-fidelity covariance matrix K_M with the elements calculated by Eqs. (36) and (37). Thereafter, we infer the hyperparameters via solving the min-NLML problem (9) by the gradient descent algorithm. The computational complexity of this step scales as $\mathcal{O}(N^3)$, which leads to computational problems if the training size N is large. In cases with a large training size, e.g., 10^4 or greater, the sparse approximations (Álvarez and Lawrence, 2009, 2011; Nguyen et al., 2014) can be employed to speed up the computation. This kind of approximation method uses z ($z \ll N$) inducing points to approximate the large covariance matrix K_M, which reduces the computational complexity to $\mathcal{O}(z^2 N)$. In this article, since the numerical examples we used have moderate training sizes (less than 10^3), we thus do not implement these sparse approximations.

Step 3 Prediction. Once the MFGP model has been trained on X and $\mathbf{y}$, we can use Eqs. (16) and (17) to predict the Q outputs jointly at a test point $\mathbf{x}_*$.

4. Numerical experiments

This section validates the performance of the proposed MFGP modeling approach on four examples with diverse characteristics and data structures. Among them, we consider two synthetic examples (a 1D pedagogical example and a 2D Branin example) with $Q = 2$, and then study two real-world engineering examples (a 6D Airfoil dataset and a 2D stochastic incompressible flow dataset) with $Q = 2$ and $Q = 3$, respectively.

For the purpose of comparison, the testing involves four GP modeling approaches:

- the SOGP modeling approach introduced in Section 2, which plays as a baseline for other MOGPs;

– the multi-task GP (MTGP) modeling approach presented by Bonilla et al. (2007). The MTGP approximates multiple correlated outputs jointly as

$$f_t(\mathbf{x}) = a_t u(\mathbf{x}), \quad 1 \le t \le Q, \tag{42}$$

where the correlation parameters $\{a_t\}_{t=1}^{Q}$ follow the full rank parameterization in Eq. (39). Due to the shared process $u(\mathbf{x})$ and the flexible correlation matrix A the MTGP can transfer useful knowledge across outputs;

– the multi-fidelity Co-Kriging modeling approach put forth by Kennedy and O'Hagan (2000). This model, denoted as CoGP in the article, is expressed in a recursive form as

$$f_t(\mathbf{x}) = a_{t-1} f_{t-1}(\mathbf{x}) + \delta_t(\mathbf{x}), \quad 2 \le t \le Q, \tag{43}$$

where a_{t-1} quantifies the correlation between $f_t(\mathbf{x})$ and $f_{t-1}(\mathbf{x})$, and the Gaussian process $\delta_t(\mathbf{x})$ represents the discrepancy between $f_t(\mathbf{x})$ and $f_{t-1}(\mathbf{x})$. Note that in CoGP, if the observations are noise-free (i.e., $\{\sigma_{s,t}^2\}_{t=1}^{Q} = 0$) and the training data is nested (i.e., $X_Q \subseteq X_{Q-1} \subseteq \cdots \subseteq X_1$), we can optimize the hyperparameters for each of the Q outputs individually based on the Markov property[6] $\mathrm{cov}[y_t(\mathbf{x}), y_{t-1}(\mathbf{x}')|y_{t-1}(\mathbf{x})] = 0$, $\forall \mathbf{x} \neq \mathbf{x}'$ (Forrester et al., 2007; Le Gratiet and Garnier, 2014). But the GP model generally considers a noise term, which is beneficial for better statistical properties (Gramacy and Lee, 2012), and this article handles the multi-fidelity problems with diverse data structure. Therefore, the hyperparameters in CoGP are optimized jointly by minimizing the NLML, like (Han et al., 2010);

– the proposed MFGP modeling approach.

We test the four modeling approaches with two HF data structures, i.e., the *full* HF points that fill the entire domain and the *partial* HF points lying in a subregion. The LF points keep filling the entire domain for providing a reliable LF model. For the HF training size n_h, we follow the $10d$ rule suggested by Loeppky et al. (2009), which can help build a good initial GP model in general. But for the 1D synthetic example below, we set $n_h = 5d$ since the $10d$ HF points are too many; for the 2D stochastic incompressible flow example below, we set $n_h = 5$ since the dataset only contains 30 HF data points. Besides, for the n_h HF points or the n_l LF points, we randomly generate 100 repetitions, i.e., we run each modeling approach 100 times. The randomness enlarges the diversity among the 100 repetitions, which helps comprehensively and statistically assess the capability of the modeling approaches. Finally, for multi-fidelity problems, we focus on the prediction accuracy of the HF model, which is assessed by the root mean square error (RMSE) criterion

$$\mathrm{RMSE} = \sqrt{\frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \left(f(\mathbf{x}_i) - \hat{f}(\mathbf{x}_i)\right)^2}, \tag{44}$$

where N_{test} is the number of test points. A smaller RMSE value indicates a higher prediction accuracy.

We employ the SE covariance function k_{SE} in Eq. (2) for the four modeling approaches. Regarding parameter settings, the length scales $\{l_i\}_{i=1}^{d}$ and the signal variance σ_f^2 are initialized to 0.5; the noise variances $\{\sigma_{s,t}^2\}_{t=1}^{Q}$ are initialized to 0.01; the lower triangular matrix L in Eq. (39) used in MTGP and MFGP is initialized with the diagonal elements as one and the remaining elements as zero, i.e., A is initialized as an identify matrix; finally, the parameter ρ in MFGP is initialized as 0.5 and the parameter a_{t-1} in CoGP is initialized as 1. Note that all the codes are implemented in the Matlab environment based on the GPML toolbox[7] and the MTGP toolbox.[8] To obtain the hyperparameters, we use the *minimize* function in the GPML toolbox to solve the min-NLML problem (9). Rasmussen and Williams (2006) pointed out that optimization over hyperparameters is a non-convex problem, and every local optimum corresponds to a particular interpretation of the training data.

4.1. A pedagogical example

As shown in Fig. 1, this 1D pedagogical example has two levels of fidelity, with f_h and f_l respectively expressed as

$$f_h(x) = 5x^2 sin(12x), \qquad x \in [0, 1], \tag{45}$$

$$f_l(x) = f_h(x) + (x^3 - 0.5)sin(3x - 0.5) + 4cos(2x), \qquad x \in [0, 1]. \tag{46}$$

It is found that f_l is a transformation of f_h including a nonlinear discrepancy. Besides, f_h and f_l show a similar trend in $[0.5, 1.0]$, but they have different trends in $[0.0, 0.5]$, see Fig. 1.

We below test the four modeling approaches for the 1D example with full and partial HF points, respectively. In the testing, we use the Matlab built-in routine *lhsdesign* to generate $n_h = 5$ full or partial HF points, and $n_l = 12$ LF points; we further vary n_h from 4 to 6 to see the effect of HF training size. Besides, the HF or LF training set has 100 repetitions for a comprehensive comparison. Finally, the RMSE value of the HF model built by each of the four modeling approaches is estimated using $N_{test} = 100$ test points.

4.1.1. Full HF points

Fig. 4(a) shows the boxplots of the RMSE values of different modeling approaches over 100 runs using $n_h = 5$ full HF points and $n_l = 12$ LF points on the 1D example. These boxplots describe how the RMSE values of the HF predictions vary over different training sets. The bottom and top of each box are the lower and upper quartile values of the RMSEs, the interior line represents the median RMSE value, the square represents the average RMSE value, the broken line (whiskers) extended from the end of the box represents the extent of the remaining data relative to the lower and upper quartiles, the maximum whisker length is 1.5 times the interquartile range, and finally the + symbols represent the outliers that beyond the limit of the broken lines.

Particularly, Fig. 5 depicts the illustrative examples of the four modeling approaches obtained from the above numerical experiments with 5 full HF points. The circles represent the HF training points, and the blue shadow represents 95% confidence interval of the predictions.

From the results in Figs. 4(a) and 5, it is observed that compared to the SOGP, all the three MOGPs are capable of transferring useful knowledge from f_l to significantly enhance the modeling of f_h. Among the three MOGPs, the MTGP model in Eq. (42) employs a pure global trend process to learn the profile of f_l for modeling f_h. Therefore, as shown in Fig. 5(b), the MTGP is able to improve over modeling f_h individually, but yielding large prediction errors in $[0.0, 0.5]$ where f_h and f_l have different trends. Compared to the MTGP, due to the additional discrepancy process in Eq. (43), the CoGP extracts more correlated information from f_l to improve the modeling of f_h. Though providing more accurate predictions more often than the MTGP, the CoGP is highly sensitive to the data structure as observed from the poor predictions in some runs for this example. In contrast, the proposed MFGP model in Eq. (34) extracts both the common and output-specific features from the LF residual information and uses them dynamically via the weight parameter ρ for different training sets. As a result, it provides the most accurate and robust predictions on this 1D example.

Finally, to investigate the impact of HF training size n_h, Fig. 4(b) shows the RMSE values of the four modeling approaches with n_h respectively being 4, 5 and 6. For illustrative purposes, this figure uses the compact formating for the box plots wherein the circles represent the median RMSE values and the open symbols represent the average RMSE values. It is observed that these modeling approaches generally perform better with the increase of n_h. Among the three MOGPs, the MFGP always provides the best HF predictions. Besides, the performance

[6] This property means that given $y_{t-1}(\mathbf{x})$, no more can be learnt about $y_t(\mathbf{x})$ from any other $y_{t-1}(\mathbf{x}')$ for $\mathbf{x} \neq \mathbf{x}'$.

[7] http://www.gaussianprocess.org/gpml/code/matlab/doc/.

[8] http://www.robots.ox.ac.uk/~davidc/publications_MTGP.php.

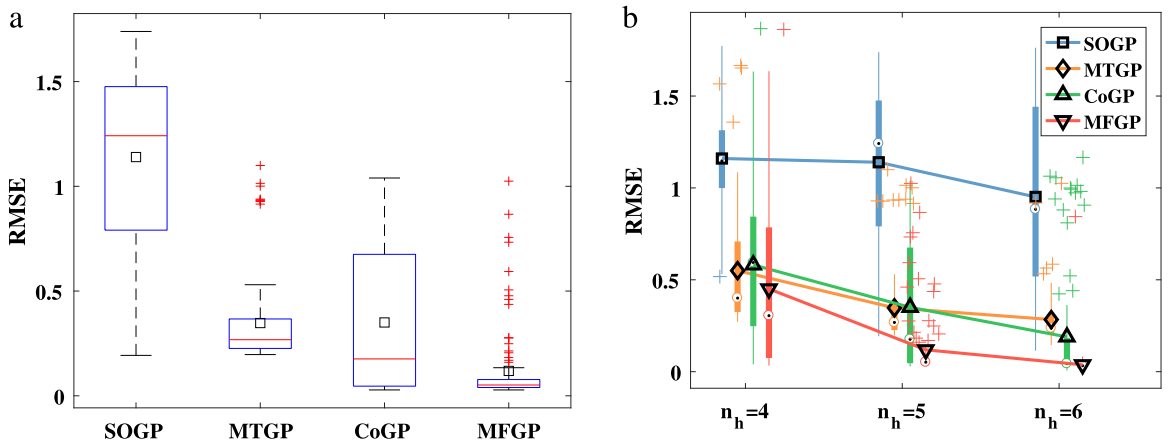

Fig. 4. The RMSE values of the four modeling approaches over 100 runs on the 1D example with full HF points. (a) The modeling results with 5 full HF points. (b) The modeling results with 4, 5 and 6 full HF points, respectively.

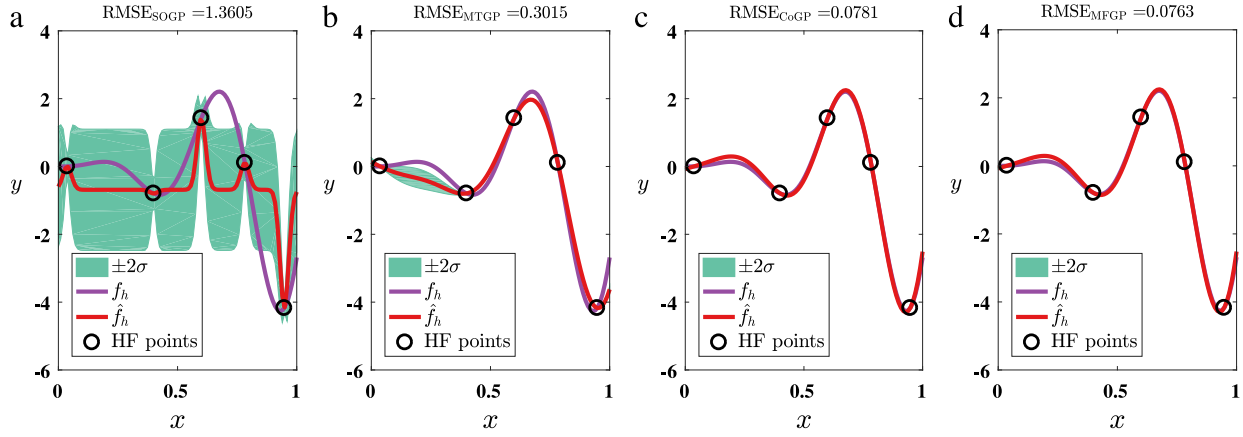

Fig. 5. The illustrative examples of (a) SOGP, (b) MTGP, (c) CoGP and (d) MFGP on the 1D example with 5 full HF points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of CoGP is sensitive to the HF training sets for the 1D example. For instance, when $n_h = 6$, the median RMSE value of CoGP over 100 runs is similar to that of MFGP. But due to the poor results in some runs, the average RMSE value of CoGP is larger than that of MFGP instead.

4.1.2. Partial HF points

This section explores the performance of different modeling approaches in cases where the HF points only exist in a subregion. In these cases, we need to extrapolate the HF predictions in the remaining regions with no training points, which is a non-trivial task. For the 1D example, since the trends of f_h and f_l are similar in $[0.5, 1.0]$ but different in $[0.0, 0.5]$, we set two cases. Case I has 5 HF points in $[0.0, 0.5]$, while Case II has 5 HF points in $[0.5, 1.0]$.

Fig. 6(a) and (c) depict the RMSE values of the four modeling approaches over 100 runs with 5 partial HF points for Case I and Case II, respectively. For Case I, all the MOGPs can improve over SOGP, and the proposed MFGP provides the best performance. For Case II, all the MOGPs except MTGP can improve over SOGP, and MFGP and CoGP yield a similar performance. Compared to the results in Fig. 4(a), it is observed that the partial HF points indeed make the modeling harder, since all the approaches here have poorer predictions with the same number of HF points.

Fig. 7 shows the illustrative examples of the four modeling approaches with 5 partial HF points for Case I and Case II, respectively. It is observed that since the MTGP contains a shared process, it provides good predictions for Case I wherein f_h and f_l have similar features in $[0.5, 1.0]$. But due to the ignorance of the residual information, the MTGP provides poor predictions for Case II wherein f_h and f_l have different features in $[0.0, 0.5]$. For example, as shown in Fig. 7(b), the MTGP learns the decreasing trend of f_l to model f_h, which actually has a slightly rising trend.

As for the CoGP model, it captures the right trend of f_h but the predictions are a bit far away from the actual f_h values for Case I. This is because with no HF points in $[0.5, 1.0]$, the discrepancy process in the CoGP model is hard to learn the nonlinear discrepancy between f_h and f_l well. For Case II, compared to the MTGP, the additional discrepancy process enables the CoGP to capture the right trend of f_h in $[0.0, 0.5]$.

On the contrary, rather than using a discrepancy process, the proposed MFGP model employs a more flexible residual formulation that learns both the shared and output-specific residual features from f_l. Hence, as shown in Fig. 7 , the MFGP can extract more LF information to provide predictions closer to f_h, especially for Case I.

Additionally, Fig. 6(b) and (d) investigate the impact of n_h on the performance of the four modeling approaches for Case I and Case II, respectively. Different from the results in Fig. 4(b), it is observed that the SOGP is hard to improve the predictions with the increase of n_h when using the partial HF points. This is because more HF points in a subregion gives more constraints to the model, which may be in conflict with the exact HF features in the remaining region. Among the three MOGPs, with the increase of n_h, the MFGP is capable of improving over SOGP, especially for Case I; the CoGP performs slightly poorer with the increase of n_h for Case I; while the MTGP even performs poorer than the SOGP with the increase of n_h for Case II.

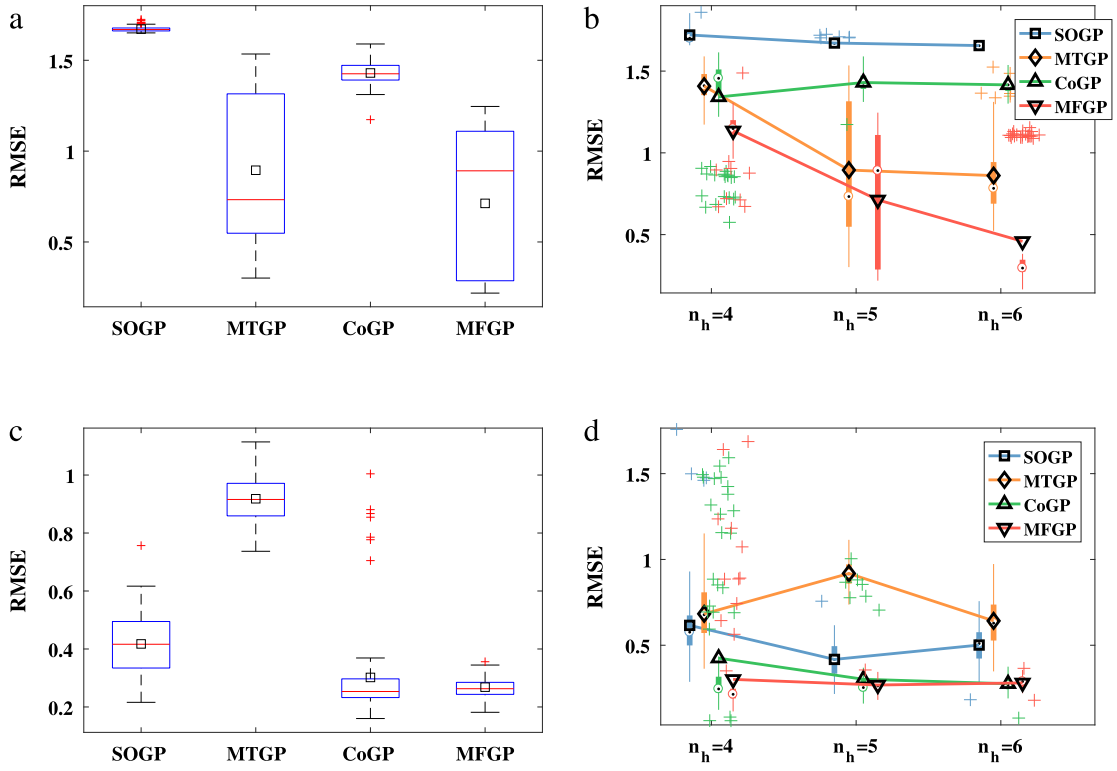

Fig. 6. The RMSE values of the four modeling approaches over 100 runs on the 1D example with partial HF points. (a) The modeling results with 5 partial HF points in [0.0, 0.5]. (b) The modeling results with 4, 5 and 6 partial HF points, respectively, in [0.0, 0.5]. (c) The modeling results with 5 partial HF points in [0.5, 1.0]. (d) The modeling results with 4, 5 and 6 partial HF points, respectively, in [0.5, 1.0].

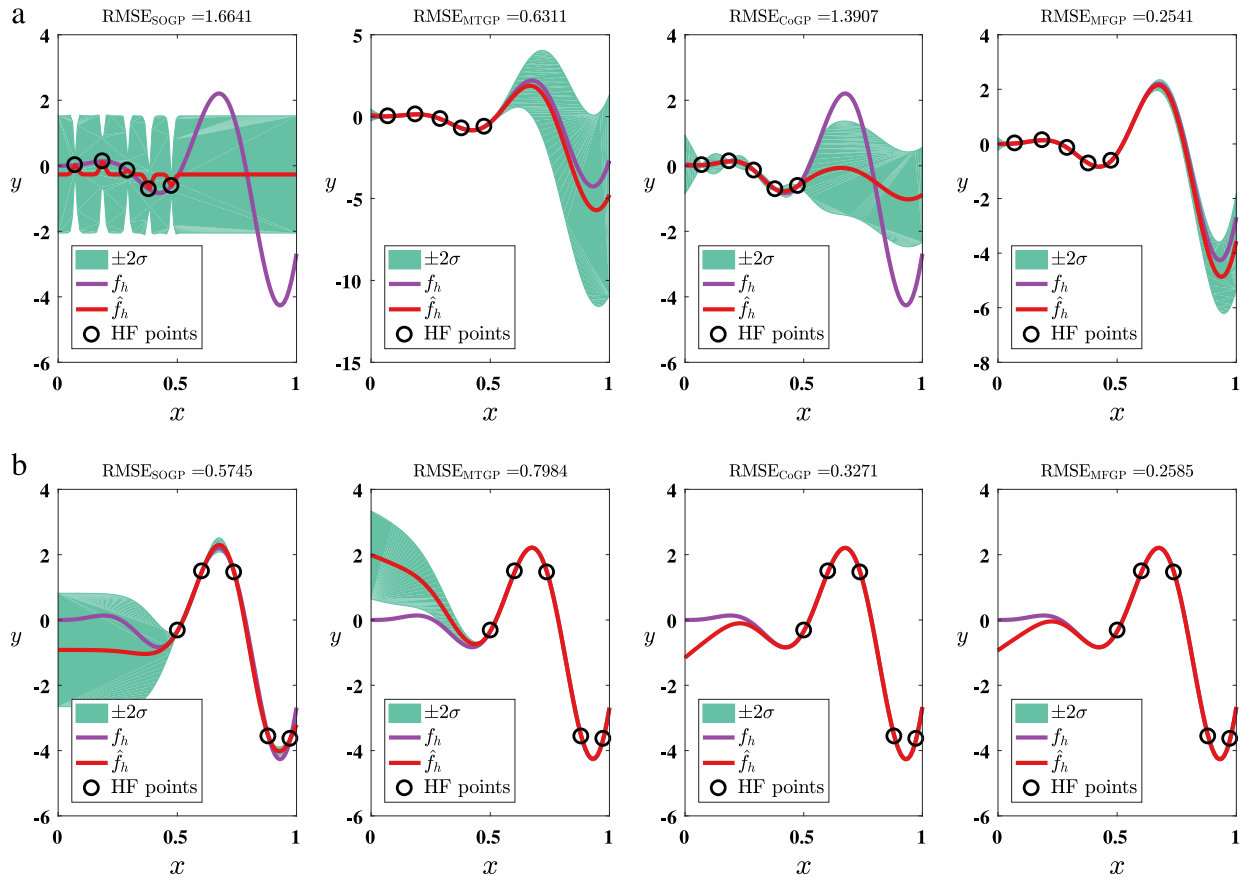

Fig. 7. The illustrative examples of the four modeling approaches with 5 partial HF points for (a) Case I and (b) Case II, respectively.

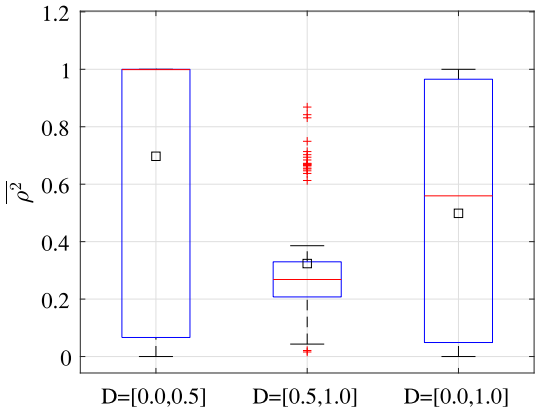

Fig. 8. The $\overline{\rho^2}$ values of MFGP varying over 100 runs on the 1D example with 5 HF points in $D = [0.0, 0.5]$, $D = [0.5, 1.0]$ and $D = [0.0, 1.0]$, respectively.

4.1.3. Impact of the weight parameter ρ

The above comprehensive results on the 1D example reveal that the proposed MFGP is very promising for MFM problems. It is found that the parameter ρ has a great impact on the performance of MFGP, since it determines the mode of residual information utilization. As shown in Eqs. (32) and (33), the parameter $\overline{\rho^2}$ represents the relative importance of the shared residual process over the output-specific residual process.

Fig. 8 shows the $\overline{\rho^2}$ values of MFGP varying over 100 runs on the 1D example with 5 HF points in $D = [0.0, 0.5]$, $D = [0.5, 1.0]$ and $D = [0.0, 1.0]$, respectively. The results offer the following findings:

- For each case, the $\overline{\rho^2}$ values over 100 runs vary from 0 to 1. It indicates that $\overline{\rho^2}$ is a data-driven parameter that enables the MFGP to utilize the shared and output-specific residual information dynamically from different training sets;
- For the case with 5 HF points in $D = [0.0, 0.5]$, the MFGP model needs to extrapolate the predictions in $[0.5, 1.0]$ where f_h and f_l are highly correlated. Hence, it assigns a $\overline{\rho^2}$ value close to 1 in many runs to fully extract the similar LF residual information in that region;
- In contrast, for the case with 5 HF points in $D = [0.5, 1.0]$, the MFGP model needs to extrapolate the predictions in $[0.0, 0.5]$ where the trends of f_h and f_l are different. Hence, MFGP prefers using more output-specific features by assigning a small $\overline{\rho^2}$ value less than 0.4 in most runs;
- Finally, for the case with 5 HF points in the entire space, the MFGP model uses different $\overline{\rho^2}$ values with a median value close to 0.5 for different training sets so as to synergize the diverse types of information.

In short, the dynamic $\overline{\rho^2}$ values over 100 runs as inferred from the various data structures enable the proposed MFGP to extract the LF information effectively when modeling f_h.

4.2. The Branin example

For the 2D Branin example in Fig. 9, f_h is the original non-stationary Branin function, while f_l is obtained by complex transformations of the original Branin function, including nonlinear scaling and shift in phase and amplitude. The expressions of f_h and f_l defined in $D \in [-5, 10] \times [0, 15]$ are respectively given as

$$f_h(\mathbf{x}) = (x_2 - \frac{5.1}{4\pi}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})cos(x_1) + 10, \quad (47)$$

$$f_l(\mathbf{x}) = 10\sqrt{f_h(\mathbf{x} - 2)} + 2(x_1 - 2.5) - 3(3x_2 - 7) - 1. \quad (48)$$

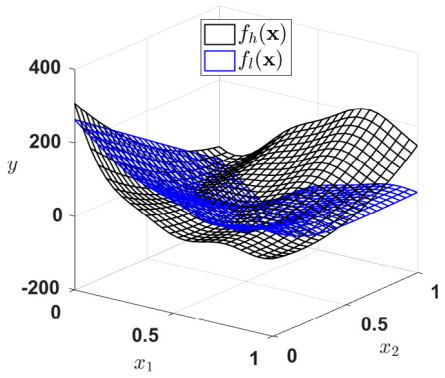

Fig. 9. The HF and LF functions of the Branin example.

Note that in Fig. 9 and the numerical experiments below, we have normalized the design space D to $[0.0, 1.0]^2$. Compared to the 1D example, the Branin example has a more complex discrepancy between f_h and f_l.

In the testing below, we use the function *lhsdesign* to generate $n_h = 20$ full or partial HF points, and $n_l = 100$ LF points; we further vary n_h from 10 to 30 to see the effect of HF training size, and vary the size of the available space that the HF points fall into to see the modeling performance. Similarly, the HF or LF training set has 100 repetitions for a comprehensive comparison. Besides, the RMSE value of the HF predictions provided by each of the four modeling approaches is estimated using $N_{test} = 5000$ test points.

4.2.1. Full HF points

Fig. 10(a) shows the RMSE values of different modeling approaches over 100 runs on the Branin example with $n_h = 20$ full HF points and $n_l = 100$ LF points. Besides, Fig. 11 depicts the illustrative examples of the four modeling approaches on the Branin example with 20 full HF points. The results obtained reveal that the proposed MFGP approach outperforms the other approaches. For instance, the predictions of the MFGP model in Fig. 11(d) are closer to f_h than those of the other models. Besides, because of the highly nonlinear discrepancy between f_h and f_l, which is hard to capture, the CoGP cannot significantly improve over the SOGP. More seriously, the MTGP performs poorer than the SOGP without considering the residual information. For instance, as shown in Fig. 11(b), the MTGP model yields large prediction errors in the top-right region.

Additionally, Fig. 10(b) investigates the impact of n_h on the performance of the four modeling approaches on the Branin example. It is observed that with the increase of n_h, the predictions of all the models have been improved. Among them, the MFGP model produces much better predictions with $n_h = 30$. The other two MOGPs, especially the MTGP model, still cannot improve over the SOGP model even with $n_h = 30$.

4.2.2. Partial HF points

Here we explore the performance of different modeling approaches on the Branin example with partial HF points. In the testing phase, the partial points are sampled in the subregion $[0.5, 1.0] \times [0.0, 1.0]$ because the responses of f_h in the remaining region are hard to predict (see Fig. 9), which is appropriate for assessing the performance of different MOGPs.

Fig. 12(a) depicts the RMSE values of different modeling approaches on the Branin example with 20 partial HF points in the subregion $[0.5, 1.0] \times [0.0, 1.0]$. Besides, Fig. 13 depicts the illustrative examples of the four modeling approaches on the Branin example with 20 partial HF points. The results in Fig. 12(a) reveal that the Branin example with partial HF points is a non-trivial task to all the approaches. The

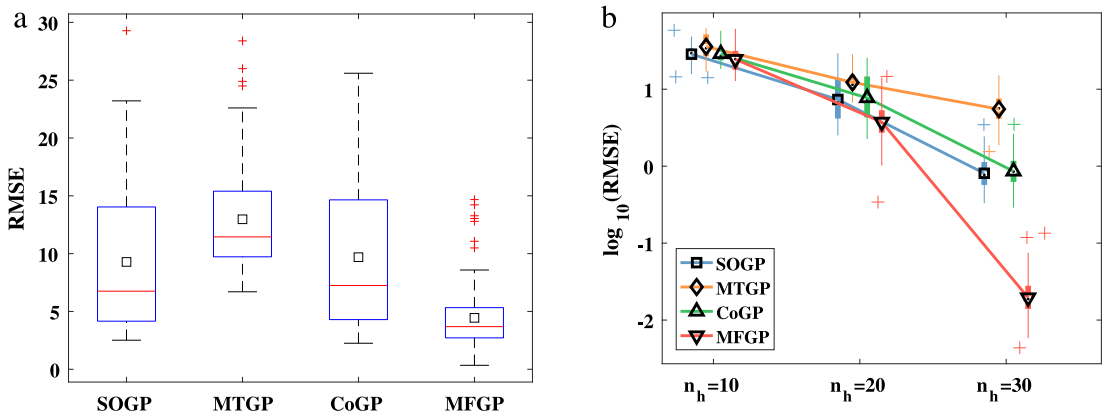

Fig. 10. The RMSE values of the four modeling approaches over 100 runs on the Branin example with full HF points. (a) The modeling results with 20 full HF points. (b) The modeling results with 10, 20 and 30 full HF points, respectively.

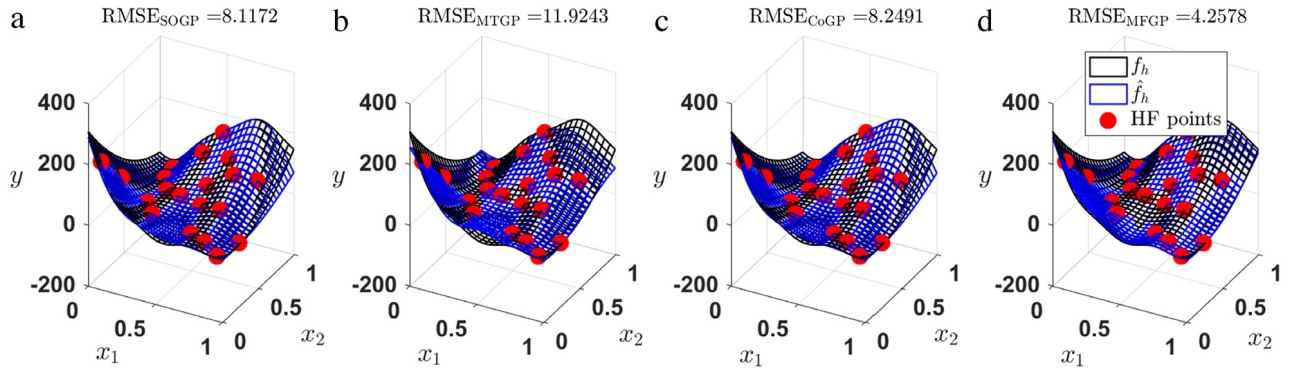

Fig. 11. The illustrative examples of (a) SOGP, (b) MTGP, (c) CoGP and (d) MFGP for the Branin example with 20 full HF points.

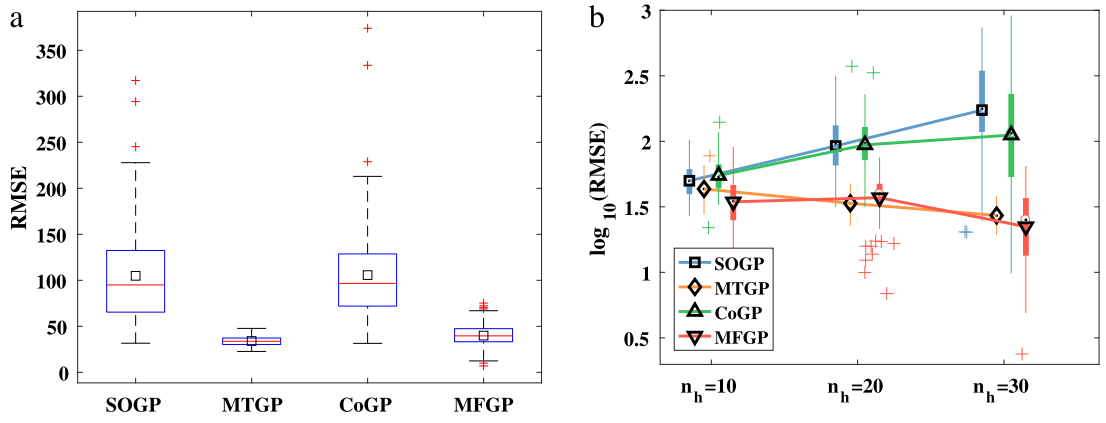

Fig. 12. The RMSE values of the four modeling approaches over 100 runs on the Branin example with partial HF points in $[0.5, 1.0] \times [0.0, 1.0]$. (a) The modeling results with 20 partial HF points. (b) The modeling results with 10, 20 and 30 partial HF points, respectively.

RMSE values obtained by these approaches here are much larger than the results using 20 full HF points in Fig. 10(a). Particularly, as shown in Fig. 13, the SOGP and CoGP models fail to provide reliable predictions in the right subregion $[0.0, 0.5] \times [0.0, 1.0]$ with no prior HF data points. The lack of HF points in the right subregion poses great difficulties for the CoGP in building a reliable discrepancy process in that space.

On the other hand, the MTGP and MFGP models, which show a similar performance, exhibit improvements over the SOGP with partial HF points. The MTGP uses the same Gaussian process to model f_h and f_l such that it can transfer the similar features of f_l to model f_h in $[0.0, 0.5] \times [0.0, 1.0]$ with no HF points, which leads to a good performance in Fig. 12(a). The proposed MFGP transfers the shared global and local features of f_l to help f_h, thus giving reliable predictions in the subregion $[0.0, 0.5] \times [0.0, 1.0]$. Besides, as shown in Fig. 13, though performing slightly poorer than MTGP in terms of RMSE, the comprehensive usage of the shared and output-specific residual information enables MFGP to capture the right trend of f_h within the region $[0.0, 0.5] \times [0.0, 1.0]$. That is the reason for the MFGP in attaining the higher accuracy than the MTGP in some runs of Fig. 12(a).

Additionally, Fig. 12(b) investigates the impact of n_h on the performance of different modeling approaches on the Branin example. It is observed that the MFGP and MTGP models generally perform better with the increase of n_h, and the MFGP model has the best performance using $n_h = 10$ and $n_h = 30$. But the SOGP and CoGP models

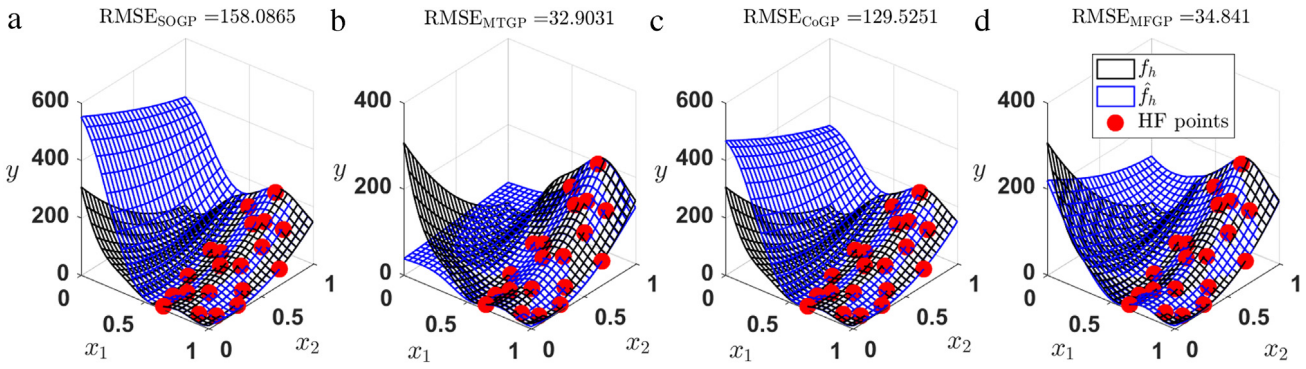

Fig. 13. The illustrative examples of (a) SOGP, (b) MTGP, (c) CoGP and (d) MFGP on the Branin example with 20 partial HF points in $[0.5, 1.0] \times [0.0, 1.0]$.

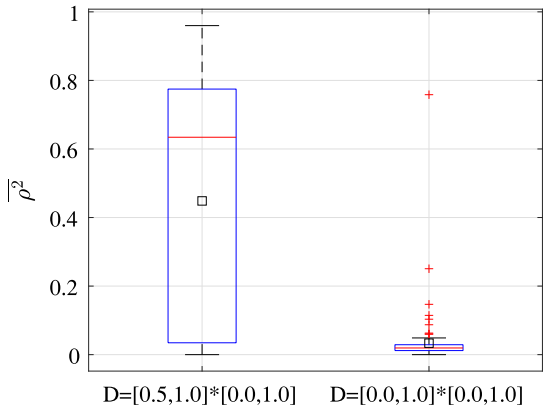

Fig. 14. The $\overline{\rho^2}$ values of MFGP varying over 100 runs on the Branin example with 20 partial and full HF points, respectively.

perform poorer with the increase of n_h. Similar phenomenon has also been observed in Fig. 6(b) and (d) for the 1D example. Their results deteriorate on the Branin example due to the higher dimensionality and the more complex discrepancy between f_h and f_l.

Finally, Fig. 14 illustrates the $\overline{\rho^2}$ values of MFGP varying over 100 runs on the Branin example with 20 partial and full HF points, respectively, so as to further study the performance of MFGP. It is observed that for the case of $D = [0.5, 1.0] \times [0.0, 1.0]$, the MFGP assigns a $\overline{\rho^2}$ value larger than 0.6 in half of the 100 runs in order to utilize more shared LF residual information as a means to improve the predictions in regions with no HF points. On the contrary, for the case of $D = [0.0, 1.0]^2$, since the profile of f_l has appropriately captured the global trend of f_h, the MFGP prefers using the output-specific residual process to model f_h by a small $\overline{\rho^2}$ value close to 0 in most runs.

4.2.3. Impact of available space size

In the above numerical experiments, the available space $D = [0.5, 1.0] \times [0.0, 1.0]$ is half of the original design space, i.e., $S_a = 0.5$. Here, our intent is to investigate the impact of available space size S_a on the performance of different modeling approaches. To this end, we run the four modeling approaches with $S_a = 0.3$ (subregion $[0.7, 1.0] \times [0.0, 1.0]$), $S_a = 0.5$ (subregion $[0.5, 1.0] \times [0.0, 1.0]$), $S_a = 0.6$ (subregion $[0.4, 1.0] \times [0.0, 1.0]$), and $S_a = 1.0$ (entire region $[0.0, 1.0]^2$), respectively. For each S_a, we generate 10, 20 and 30 HF points in the associated region, respectively. Note that the modeling results of $S_a = 1.0$ and $S_a = 0.5$ have been reported in Figs. 10 and 12, respectively.

Fig. 15 shows the average scores of different modeling approaches with the increase of available space size S_a and HF training size n_h on the Branin example. The score is defined as the normalized $1/\log_{10}$RMSE value, since the RMSE values of different approaches have considerable differences in magnitude in some circumstances. Hence, a larger circle in Fig. 15 denotes a higher prediction accuracy.

The comprehensive results obtained reveal that the increase of S_a and n_h brings about greater HF information, thus facilitating the MFGP and MTGP models in attaining more accurate predictions in general. The SOGP and CoGP models also perform better with the increase of S_a; however, they perform poorer with the increase of n_h when $S_a < 1$.

4.3. The Airfoil example

The airfoil example (Burnaev and Zaytsev, 2016) calculates the *lift* and *drag* coefficients of an airfoil under different flight conditions and geometry parameters. The original airfoil is parameterized by 52 design variables including the geometry parameters, the speed and the angle of attack. According to a dimension reduction process, the airfoil data is generated based on six most important design variables (Bernstein et al., 2011). Two solvers with different levels of fidelity were used to obtain the lift and drag coefficients. For each coefficient, the airfoil dataset contains 365 HF points and 1996 LF points. Note that in the testing below we have normalized the dataset to $[0, 1]^6$.

For both the lift and drag coefficients, we set $n_h = 60$ and $n_l = 400$, and generate 100 repetitions, each of which is randomly selected from

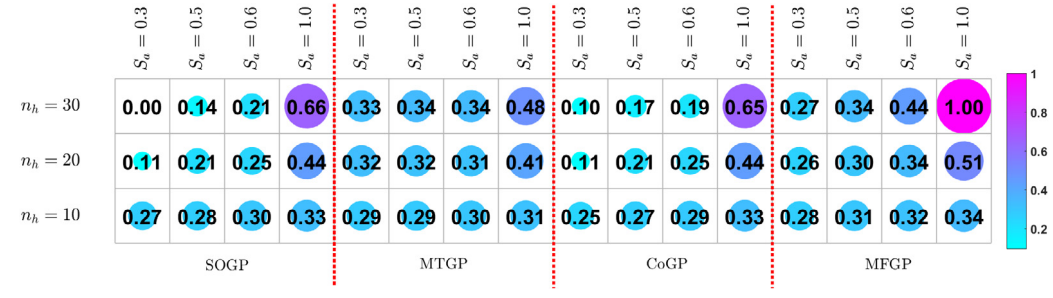

Fig. 15. The average scores of different modeling approaches on the Branin example with the increase of available space size S_a and HF training size n_h.

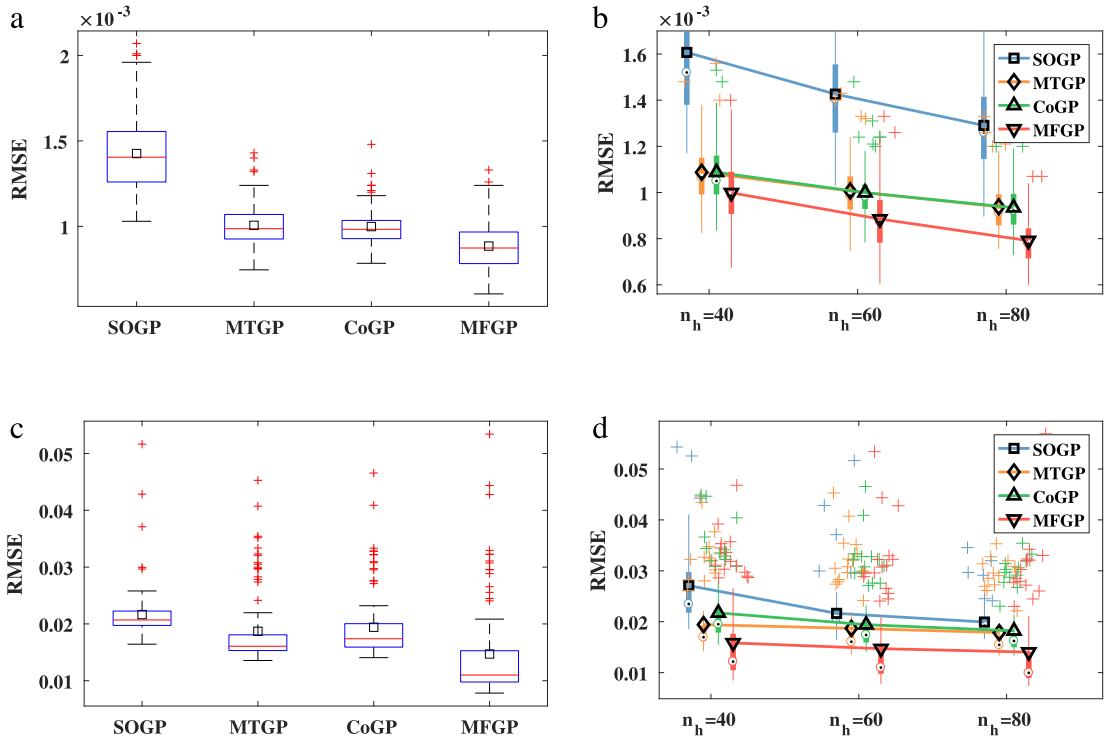

Fig. 16. The RMSE values of the four modeling approaches over 100 runs on the Airfoil example with full HF points. (a) The modeling results for the lift coefficient with 60 full HF points. (b) The modeling results for the lift coefficient with 40, 60 and 80 full HF points, respectively. (c) The modeling results for the drag coefficient with 60 full HF points. (d) The modeling results for the drag coefficient with 40, 60 and 80 full HF points, respectively.

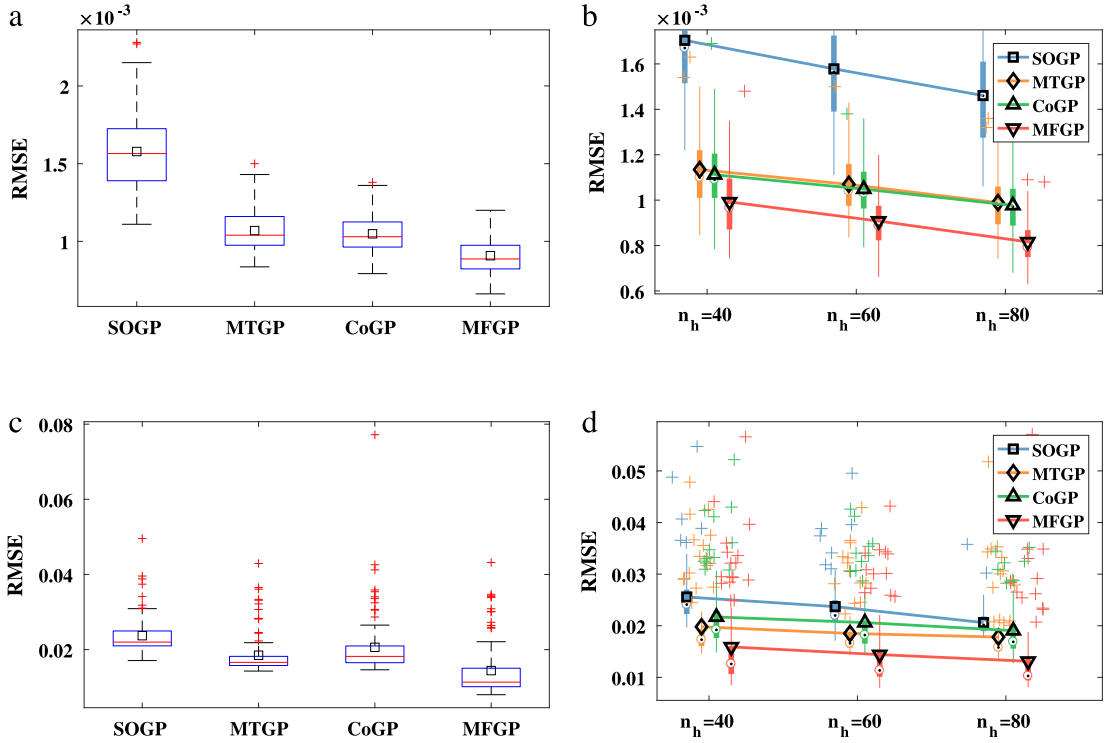

Fig. 17. The RMSE values of the four modeling approaches over 100 runs on the Airfoil example with partial HF points in $[0.5, 1.0] \times [0.0, 1.0]^5$. (a) The modeling results for the lift coefficient with 60 partial HF points. (b) The modeling results for the lift coefficient with 40, 60 and 80 partial HF points, respectively. (c) The modeling results for the drag coefficient with 60 partial HF points. (d) The modeling results for the drag coefficient with 40, 60 and 80 partial HF points, respectively.

the original dataset. Besides, we select 200 HF test points from the dataset to evaluate the RMSE value. Fig. 16(a) shows the RMSE values of different modeling approaches for the lift coefficient with 60 full HF points, and Fig. 16(b) investigates the impact of n_h on the performance of different modeling approaches for the lift coefficient. The results obtained reveal that all the MOGPs are able to improve over the SOGP;

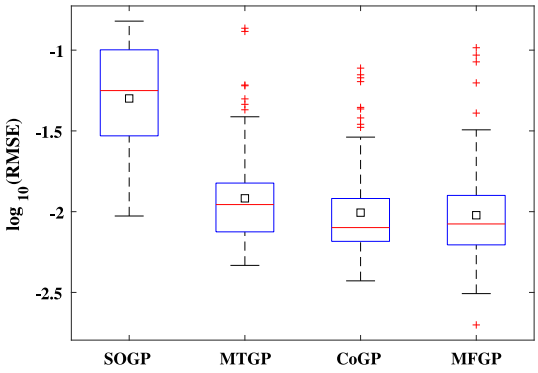

Fig. 18. The RMSE values of the four modeling approaches over 100 runs on the SIFlow example with 5 HF points, 15 MF points and 75 LF points.

the proposed MFGP has the best performance with different HF training sizes; and finally, the CoGP and MTGP perform similarly for the lift coefficient.

Fig. 16(c) shows the RMSE values of different modeling approaches for the drag coefficient with 60 full HF points, and Fig. 16(d) depicts the impact of n_h on the performance of different modeling approaches for the drag coefficient. The results obtained are similar to those for the lift coefficient. The major differences are that for the drag coefficient, (1) the modeling approaches are more sensitive (with many outliers) to the training sets; and (2) the MTGP has a slightly better performance than the CoGP.[9]

Additionally, Fig. 17(a)–(d) show the RMSE values of different modeling approaches for the lift and drag coefficients with partial HF points in $[0.5, 1.0] \times [0.0, 1.0]$[5]. Again, the MFGP provides the highest prediction accuracy in different scenarios. Besides, different from the results of the 1D example and the Branin example, it is found that the RMSE values of these modeling approaches using partial HF points on the Airfoil example are close to those using full HF points in Fig. 16. This is because for both the lift and drag coefficients, the correlation between the HF and LF solvers is higher than that of the 1D example and the Branin example.[10]

4.4. The stochastic incompressible flow example with three-level fidelity

The final real-world engineering example with three-level fidelity investigates a 2D stochastic incompressible flow (SIFlow) passing a circular cylinder under a random inflow boundary condition (Perdikaris et al., 2015). The goal of this example is to model the 0.6-superquantile risk of the base pressure coefficient $\mathcal{R}_{0.6}(C_{BP})$ with the condition of Reynolds number $Re = 100$. Because of the random process, this problem should be simulated in both physical space and probability space. In physical space, the incompressible flow field governed by the Navier–Stokes equations is solved by the spectral/hp element method (SEM) (Karniadakis and Sherwin, 2013).

The levels of fidelity of this example come from the simulations in probability space using three models of different precisions. The computational time of the HF model, known as probabilistic collocation on a tensor product grid (PCM), is about 3 times the medium fidelity (MF) model, labeled as smolyak sparse grid level-2 quadrature (SG-L2), and is about 15 times the LF model, called Monte Carlo integration (MC). The SIFlow dataset contains 30 HF points, 99 MF points and 357 LF points, which are simulated in parallel on one rack of IBM BG/Q (16 384 cores) (Perdikaris et al., 2015). Note that in the testing below we have normalized the dataset to $[0, 1]^2$.

Firstly, for the SIFlow example with three-level fidelity, we test different modeling approaches using 5 HF points, 15 MF points and 75 LF points. Each training set has 100 repetitions randomly selected from the given dataset. Besides, we select 10 HF test points from the dataset to estimate the RMSE value. Fig. 18 shows the RMSE values of the four modeling approaches over 100 runs on the SIFlow example. It is observed that all the three MOGPs are capable of transferring useful knowledge from the MF and LF outputs for enhancing the modeling of the HF output. Among them, the MFGP and CoGP perform better than the MTGP.

Suppose that the computing time of an LF simulation is 1, then the MF computing time is 5, and the HF computing time is 15. For the 5HF-15MF-75LF combination in Fig. 18, the total computational budget is 225. Besides, we know that the correlation between the HF and MF solvers is higher than that between the HF and LF solvers.[11] If we fix the HF training size as 5, an interesting question arises, i.e., how to assign the remaining computational resource to the MF and LF solvers in order to enhance the HF model as much as possible?

In response to this question, we set five different combinations of the HF-MF-LF training size under the same computational budget, and show the RMSE values of the four modeling approaches with these configurations in Fig. 19. The comparative results suggest that it is better to assign the total computational budget to the two highest fidelity levels (i.e., $n_h = 5, n_m = 30$). This may be because the correlation between the HF and MF solvers is higher, thus purely transferring knowledge from the MF results provides more benefits for the HF modeling.

[9] The p-value of statistical t-test on the RMSE values of MTGP and CoGP is 0.054.

[10] For the lift coefficient, the Pearson correlation coefficient between the HF and LF solvers is 0.90; for the drag coefficient, the Pearson correlation coefficient is 0.99.

[11] For the SIFlow example, the Pearson correlation coefficient between the HF and MF solvers is 0.9997; while the correlation coefficient between the HF and LF solvers is 0.9904.

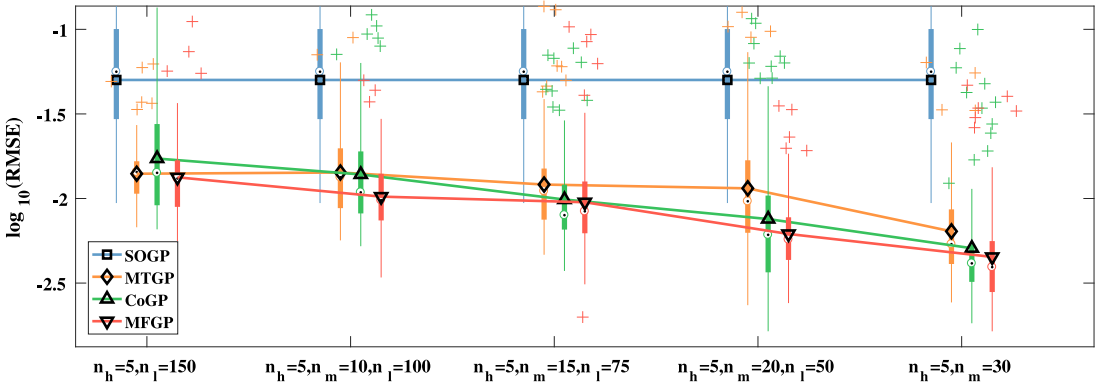

Fig. 19. The RMSE values of the four modeling approaches over 100 runs on the SIFlow example with different combinations of the HF-MF-LF training size under the same computational budget.

5. Conclusions

This article presents a novel multi-fidelity Gaussian process model for multi-fidelity regression problems with diverse data structures. Numerical results reveal that the proposed MFGP modeling approach has a promising performance in different scenarios. For different HF data structures, e.g., full HF points and partial HF points, the MFGP can effectively extract the LF information so as to enhance the modeling of the HF output. Future research efforts are needed to improve the modeling efficiency of MFGP for handling cases with "Big" data and dimensionality.

Acknowledgments

This work was conducted within the Rolls-Royce@NTU Corporate Lab with support from the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme. It is also partially supported by the Data Science and Artificial Intelligence Research Center (DSAIR) and the School of Computer Science and Engineering at Nanyang Technological University.

References

Ababou, R., Bagtzoglou, A.C., Wood, E.F., 1994. On the condition number of covariance matrices in kriging, estimation, and simulation of random fields. Math. Geol. 26 (1), 99–133.

Álvarez, M., Lawrence, N.D., 2009. Sparse convolved Gaussian processes for multi-output regression. In: NIPS, pp. 57–64.

Álvarez, M.A., Lawrence, N.D., 2011. Computationally efficient convolved multiple output gaussian processes. J. Mach. Learn. Res. 12 (May), 1459–1500.

Álvarez, M.A., Rosasco, L., Lawrence, N.D., et al., 2012. Kernels for vector-valued functions: A review. Found. Trends® Mach. Learn. 4 (3), 195–266.

Benamara, T., Breitkopf, P., Lepot, I., Sainvitu, C., 2016. Multi-fidelity extension to non-intrusive proper orthogonal decomposition based surrogates. In: ECCOMAS Congress 2016, pp. 4129–4145.

Bernstein, A., Burnaev, E., Chernova, S., Zhu, F., Qin, N., 2011. Comparison of three geometric parameterization methods and their effect on aerodynamic optimization. In: Proceedings of International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems, Eurogen, pp. 14–16.

Bilionis, I., Zabaras, N., 2012. Multi-output local Gaussian process regression: Applications to uncertainty quantification. J. Comput. Phys. 231 (17), 5718–5746.

Bonilla, E.V., Chai, K.M.A., Williams, C.K., 2007. Multi-task Gaussian process prediction. In: NIPS, Vol. 20, pp. 153–160.

Boyle, P., Frean, M.R., 2004. Dependent Gaussian processes. In: NIPS, Vol. 17, pp. 217–224.

Burnaev, E., Zaytsev, A., 2016. Minimax error of interpolation and optimal design of experiments for variable fidelity data. arXiv preprint arXiv:1610.06731.

Dürichen, R., Pimentel, M.A., Clifton, L., Schweikard, A., Clifton, D.A., 2015. Multitask Gaussian processes for multivariate physiological time-series analysis. IEEE Trans. Biomed. Eng. 62 (1), 314–322.

Fernández-Godino, M.G., Park, C., Kim, N.-H., Haftka, R.T., 2016. Review of multi-fidelity models. arXiv preprint arXiv:1609.07196.

Forrester, A.I., Sóbester, A., Keane, A.J., 2007. Multi-fidelity optimization via surrogate modelling. Proc. R. Soc. A: Math. Phys. Eng. Sci. 463 (2088), 3251–3269.

Gramacy, R.B., Lee, H.K., 2012. Cases for the nugget in modeling computer experiments. Stat. Comput. 22 (3), 713–722.

Han, Z.-H., Görtz, S., 2012. Hierarchical kriging model for variable-fidelity surrogate modeling. AIAA J. 50 (9), 1885–1896.

Han, Z.-H., Görtz, S., Zimmermann, R., 2013. Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. Aerosp. Sci. Technol. 25 (1), 177–189.

Han, Z.-H., Zimmermann, R., Goretz, S., 2010. A new cokriging method for variable-fidelity surrogate modeling of aerodynamic data. In: 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, pp. AIAA 2010–1225.

Hayashi, K., Takenouchi, T., Tomioka, R., Kashima, H., 2012. Self-measuring similarity for multi-task gaussian process. In: ICML, Vol. 27, pp. 145–154.

Hori, T., Montcho, D., Agbangla, C., Ebana, K., Futakuchi, K., Iwata, H., 2016. Multi-task Gaussian process for imputing missing data in multi-trait and multi-environment trials. Theor. Appl. Genet. 129 (11), 2101–2115.

Journel, A.G., Huijbregts, C.J., 1978. Mining Geostatistics. Academic Press.

Karniadakis, G., Sherwin, S., 2013. Spectral/hp Element Methods for Computational Fluid Dynamics. Oxford University Press.

Keane, A.J., 2012. Cokriging for robust design optimization. AIAA J. 50 (11), 2351–2364.

Kennedy, M.C., O'Hagan, A., 2000. Predicting the output from a complex computer code when fast approximations are available. Biometrika 87 (1), 1–13.

Kennedy, M.C., O'Hagan, A., 2001. Bayesian calibration of computer models. J. R. Stat. Soc. Ser. B Stat. Methodol. 63 (3), 425–464.

Kleijnen, J.P., Mehdad, E., 2014. Multivariate versus univariate Kriging metamodels for multi-response simulation models. European J. Oper. Res. 236 (2), 573–582.

Kontogiannis, S.G., Savill, A.M., Kipouros, T., 2017. A Multi-Objective Multi-Fidelity framework for global optimization. In: 58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, pp. AIAA 2017–0136.

Le Gratiet, L., Cannamela, C., 2015. Cokriging-based sequential design strategies using fast cross-validation techniques for multi-fidelity computer codes. Technometrics 57 (3), 418–427.

Le Gratiet, L., Garnier, J., 2014. Recursive co-kriging model for design of computer experiments with multiple levels of fidelity. Int. J. Uncertain. Quantif. 4 (5), 365–386.

Leen, G., Peltonen, J., Kaski, S., 2012. Focused multi-task learning in a Gaussian process framework. Mach. Learn. 89 (1–2), 157–182.

Loeppky, J.L., Sacks, J., Welch, W.J., 2009. Choosing the sample size of a computer experiment: A practical guide. Technometrics 51 (4), 366–376.

Lophaven, S.N., Nielsen, H.B., Søndergaard, J., 2002. Aspects of the matlab toolbox DACE. Technical Report Informatics and Mathematical Modelling, Technical University of Denmark, DTU.

Myers, D.E., 1982. Matrix formulation of co-kriging. Math. Geol. 14 (3), 249–257.

Neal, R.M., 1997. Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. arXiv preprint physics/9701026.

Nguyen, T.V., Bonilla, E.V., et al., 2014. Collaborative multi-output Gaussian processes. In: UAI, pp. 643–652.

Osborne, M.A., Roberts, S.J., Rogers, A., Jennings, N.R., 2012. Real-time information processing of environmental sensor network data using bayesian gaussian processes. ACM Trans. Sensor Netw. 9 (1), Article No. 1.

Park, C., Haftka, R.T., Kim, N.H., 2017. Remarks on multi-fidelity surrogates. Struct. Multidiscip. Optim. 55 (3), 1029–1050.

Perdikaris, P., Karniadakis, G.E., 2016. Model inversion via multi-fidelity Bayesian optimization: a new paradigm for parameter estimation in haemodynamics, and beyond. J. R. Soc. Interface 13 (118), 20151107.

Perdikaris, P., Raissi, M., Damianou, A., Lawrence, N., Karniadakis, G., 2017. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. Proc. R. Soc. A: Math. Phys. Eng. Sci. 473 (2198), 20160751.

Perdikaris, P., Venturi, D., Royset, J., Karniadakis, G., 2015. Multi-fidelity modelling via recursive co-kriging and Gaussian–Markov random fields. Proc. R. Soc. A: Math. Phys. Eng. Sci. 471 (2179), 20150018.

Qian, P.Z., 2009. Nested Latin hypercube designs. Biometrika 96 (4), 957–970.

Qian, P.Z., Wu, C.J., 2008. Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments. Technometrics 50 (2), 192–204.

Rakitsch, B., Lippert, C., Borgwardt, K., Stegle, O., 2013. It is all in the noise: Efficient multi-task Gaussian process inference with structured residuals. In: NIPS, pp. 1466–1474.

Rasmussen, C.E., Williams, C.K.I., 2006. Gaussian Processes for Machine Learning. MIT Press.

Seeger, M., Teh, Y.-W., Jordan, M., 2005. Semiparametric latent factor models. Technical Report EPFL-REPORT-161465.

Toal, D.J., 2015. Some considerations regarding the use of multi-fidelity Kriging in the construction of surrogate models. Struct. Multidiscip. Optim. 51 (6), 1223–1245.

Ulaganathan, S., Couckuyt, I., Ferranti, F., Laermans, E., Dhaene, T., 2015. Performance study of multi-fidelity gradient enhanced kriging. Struct. Multidiscip. Optim. 51 (5), 1017–1033.

Vargas-Guzmán, J., Warrick, A., Myers, D., 2002. Coregionalization by linear combination of nonorthogonal components. Math. Geol. 34 (4), 405–419.

Ver Hoef, J.M., Barry, R.P., 1998. Constructing and fitting models for cokriging and multivariable spatial prediction. J. Statist. Plann. Inference 69 (2), 275–294.