AIAA JOURNAL
Vol. 60, No. 1, January 2022

Gradient-Enhanced Multi-Output Gaussian Process Model for Simulation-Based Engineering Design

Quan Lin,* Jiexiang Hu,† Lili Zhang,‡ Peng Jin,§ Yuansheng Cheng,¶ and Qi Zhou**
Huazhong University of Science & Technology, 430074 Wuhan, People's Republic of China

https://doi.org/10.2514/1.J060728

The multi-output Gaussian process model has shown a promising way to deal with multiple related outputs. It can capture some useful information across outputs so as to provide more accurate predictions than simply modeling these outputs separately. If incorporating gradient formation into the modeling construction, the accuracy of the model can be further improved. The main original contribution of this work is to propose a multi-output Gaussian process model assisted by gradient information, which can enhance the prediction accuracies of multiple outputs simultaneously. The observed response values, as well as the gradient information of all the outputs, are incorporated in the covariance matrix. In such a structure, not only the observed responses but also the correlation information across different outputs can be fully used. The proposed model is demonstrated with two analytical examples and used for modeling aerodynamic coefficients of a NACA 0012 airfoil. Three other existing Gaussian-process-based models are also tested to compare with the proposed model. Results show that the proposed model is promising when it comes to the problems with multiple related outputs and the prediction accuracy can be enhanced with the help of gradient information.

Nomenclature

a	=	hyperparameters in k^t
$\boldsymbol{f}$	=	vector of T outputs
$\mathcal{GP}$	=	Gaussian process
K_{GM}	=	training set covariances in the proposed model
K_M	=	training set covariances in the multitask Gaussian process model
K_*	=	training-test set covariances in the multitask Gaussian process model
K_{**}	=	test set covariances in the multitask Gaussian process model
K^x	=	correlation matrix over inputs in the proposed model
k^t	=	correlation matrix between tasks
k^x	=	correlation matrix over inputs in the multitask Gaussian process model
L	=	triangular matrix after Cholesky decomposition of k^t
l^2	=	characteristic length scales
m	=	number of design variables
N	=	total number of design sites of T outputs
N_{test}	=	total number of test points
n	=	number of design sites
n_p	=	total number of hyperparameters
P	=	matrix containing the characteristic length scales
S	=	sampling sites
w	=	number of parameters in k^t
X	=	collection of sampling sites of T outputs
X_i	=	sampling sites of the ith output
$\boldsymbol{x}$	=	design variables
$\boldsymbol{x}_*$	=	new test point
Y	=	matrix of observed function values
$\boldsymbol{y}$	=	vector of observed function values of T outputs
$\boldsymbol{y}_i$	=	vector of observed function values of the ith output
δ^2	=	gradient errors
ε_t	=	Gaussian noise of the tth output
Σ_s	=	diagonal matrix of Gaussian noise
Σ_M	=	noise matrix in the multitask Gaussian process model
$\sigma_{s,t}^2$	=	variance of the Gaussian noise of the tth output
σ_t^2	=	process variance of the tth output
υ^2	=	observation errors
$\otimes$	=	Kronecker product

Subscripts

s, t	=	index $\in [1, T]$, referring to the sth or tth output
u, v	=	index $\in [1, m]$, referring to the uth or vth dimension

Superscripts

$(i), (j)$	=	index $\in [1, n]$, referring to the ith or jth sampling site
$\sim$	=	quantities associated with the proposed model
$\hat{}$	=	approximated value

I. Introduction

COMPUTATIONAL simulation models (e.g., computational fluid dynamics [CFD] models and finite-element [FE] analysis models) have been widely used in engineering design and optimization to replace the expensive physical experiments. However, running these simulation models to evaluate a mass of design alternatives when exploring the design space for optimal solutions may still be time-consuming. A method to relieve the computational burden is to approximate the simulation model using surrogate models, e.g., Gaussian process (GP) model (also known as *kriging*) [1], radial basis function (RBF) model [2], and support vector regression (SVR) model [3]. Among these surrogate models, the GP model can not only predict the responses of the unobserved points but also provide the predicted variances. In consequence, it has gained widespread applications in many fields, e.g., machine learning [4], stress-based topology optimization [5], and model validation [6].

In spite of the popularity in different fields, the typical single-output Gaussian process (SOGP) can just model the outputs separately when it comes to multiple outputs. If these outputs have correlations in a way, modeling these outputs separately may lose some useful information. What is more, there have arisen lots of engineering problems with multiple dependent outputs, e.g., physiological time-series analysis [7], uncertainty propagation of frequency response functions [8], and

Received 23 March 2021; revision received 13 May 2021; accepted for publication 17 May 2021; published online 30 July 2021.

*Ph.D. Candidate, School of Aerospace Engineering.
†Postdoctor, School of Aerospace Engineering.
‡Ph.D. Candidate, The State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering.
§Associate Professor, School of Aerospace Engineering.
¶Professor, School of Naval Architecture and Ocean Engineering.
**Associate Professor, School of Aerospace Engineering; qizhouhust@gmail.com. Member AIAA (Corresponding Author).

multi-response surfaces for airfoil design [9]. The multi-output Gaussian process (MOGP) model has been proposed to deal with several coupled outputs, which can capture the correlation across outputs. A challenge in the MOGP modeling is to build a positive semidefinite covariance matrix that is capable of catching the correlations across outputs. Such a structure can help transfer useful information across outputs so as to improve the prediction accuracy of the model. In the MOGP modeling, there are mainly two different types of covariance structures, separable and nonseparable. If the covariance function can be constructed as a product of a covariance function for the input space alone and a covariance function only for the correlations across outputs, this type of covariance function is called a separable covariance function. The most widely used decoupled inputs and outputs separable covariance structure is known as the linear model of coregionalization (LMC) [10]. Another well-known model is the intrinsic coregionalization model (ICM) [11], which is a simplified version of the LMC. There are different applications in the machine learning field based on the LMC or ICM structure, such as the semiparametric latent factor model (SLFM) [12] and the multitask Gaussian process (MTGP) prediction [13]. As for the nonseparable covariance functions, a popular method to build this type of covariance functions is to use convolution processes (CPs). It is a nonseparable structure that builds the kernel functions by convolving a given process with smoothing kernels. Boyle and Frean [14] formulated a nonseparable kernel through a white Gaussian noise convolved with smoothing kernels. Alvarez and Lawrence [15] generalized this method to convolve general Gaussian processes with smoothing kernels. Some reviews and comparison studies regarding the MOGP modeling can be found in [16,17].

The performance of GP-based models is greatly dependent on the ability of the covariance function to capture the actual features of the function to be modeled. Hence, incorporating auxiliary information such as gradient information into the covariance function can help to enhance the accuracy of the surrogate model. Since some efficient gradient evaluation methods, such as adjoint methods [18] and automatic differentiation [19], have been successfully applied in the field of aerodynamics [20–22], exploiting gradient information as additional training data has become more attractive. There are two different ways when incorporating gradient information into the surrogate model. One is to add gradient information directly into the modeling equation system. Morris et al. [23] first proposed an extension of kriging using gradient information to construct the surrogate model, which is known as the direct gradient enhanced kriging (GEK) [23,24]. Ulaganathan et al. [25] studied the performances of direct GEK, pointing that using additional gradients information can significantly reduce the required number of expensive simulations. de Baar et al. [26] derived GEK using Bayesian theorem and found that the robustness of the GEK model was improved when considering observed response errors and gradient errors. Han et al. [27] proposed a novel formulation of gradient-enhanced kriging for high-dimensional problems, which can reduce the cost of model training dramatically. Another way to construct the GEK model is to combine the estimated nearby function values by first-order Taylor approximation with observed response values to build a standard kriging model, called indirect GEK [28,29]. Zimmermann [30] compared the performance of the direct GEK and the indirect GEK, which exposed the connection between the two types of GEK models, and both are widely used in applications. Liu and Batill [31] presented an approach to determine the step size for the design variables when using Taylor approximation in the indirect GEK. For more details about the applications of gradient-enhanced surrogates, readers can refer to the review paper [32].

However, to the best of our knowledge, there is scarce research about using gradient information to improve the accuracy of the model in the MOGP modeling. The existing researches mainly focus on one single-output situation enhanced by gradient information. Therefore, a gradient-enhanced MOGP (GEMOGP) model is proposed in this paper to enhance the predictions of all the outputs simultaneously. In the GEMOGP model, not only the correlations across outputs can be captured, but also the gradient information can be fully used.

The remainder of this paper is organized as follows: Sec. II gives a brief overview of the framework of the MTGP model. Section III presents the detailed formulation of the proposed GEMOGP model, including the predictor and relevant uncertainties, the selection of the correlation models, and parameter estimation. Section IV demonstrates the performance of the proposed GEMOGP model on two analytical examples and the prediction of aerodynamic coefficients for the airfoil NACA 0012. Finally, conclusions and future work are given in Sec. V.

II. Multitask Gaussian Process Framework

The MTGP model was first proposed by Bonilla et al. [13] for multitask learning in the context of a GP prior. Some terminology should be introduced first. If each output shares the same training set, it is known as the isotopic data. While if each output has different training points, it is called the heterotopic data [33]. The MTGP model can be applied to both types of training data. For convenience, the isotopic data are used to demonstrate in the following statements.

In the multi-output scenario, if the number of the outputs is T, given a set of n design sites $S = [\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}]^T$ with $\boldsymbol{x}^{(j)} \in R^m$, the corresponding observations are $Y = [\boldsymbol{y}_1, \cdots \boldsymbol{y}_T]$ with $\boldsymbol{y}_t \in R^n$. In the MTGP structure, let X be the collection of all the sampling locations of the T outputs:

$$X = \left\{X_1^T, \ldots, X_T^T\right\}^T \tag{1}$$

where $X_1 = \ldots = X_T = S$ for the isotopic data, and $\boldsymbol{y}$ be the vector of the relevant observations:

$$\boldsymbol{y} = \left\{\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_T^T\right\}^T \tag{2}$$

In the MOGP framework, the T outputs $f(x) = \{f_1, \ldots, f_T\}^T$ are assumed to follow a Gaussian process:

$$\boldsymbol{f}(\boldsymbol{x}) \sim \mathcal{GP}(\boldsymbol{0}, K_M(\boldsymbol{x}, \boldsymbol{x}')) \tag{3}$$

The covariance between two different outputs can be formulated as [13]

$$K_M(\boldsymbol{x}_t, \boldsymbol{x}_{t'}') = k^t(t, t') \bullet k^x(\boldsymbol{x}, \boldsymbol{x}') \tag{4}$$

where k^t is a positive semidefinite (PSD) matrix independent of the input space, which represents the correlation or similarity between tasks, and k^x is a standard covariance function over inputs, in which a stationary covariance function is typically used [4]. The full covariance matrix for the training data is formulated as

$$K_M(X, X) = k^t \otimes k^x \tag{5}$$

where $\otimes$ is the Kronecker product, k^t has a size of $T \times T$, k^x is an $n \times n$ matrix of the inputs, and K_M is a matrix with $nT \times nT$.

In the MTGP model, it is assumed that

$$y_t(\boldsymbol{x}) = f_t(\boldsymbol{x}) + \varepsilon_t \tag{6}$$

where $\varepsilon_t \sim \mathcal{N}(0, \sigma_{s,t}^2)$ is an additive independent and identically distributed (i.i.d) Gaussian noise of the tth output. Note that the consideration of the noise can not only help to improve the robustness of the matrix, but also transfer information across the outputs [34]. It should be pointed out that, for the isotopic training data, there will be a cancellation of intertask transfer if noiseless observations ($\varepsilon_t = 0$) are considered for all the outputs [13]. The likelihood function can be formulated as

$$p(\boldsymbol{y}|\hat{\boldsymbol{f}}, \boldsymbol{x}, \Sigma_s) = \mathcal{N}(\hat{\boldsymbol{f}}(\boldsymbol{x}), \Sigma_s) \tag{7}$$

where Σ_s is a $T \times T$ diagonal matrix with the elements $\{\sigma_{s,t}^2\}_{t=1,\ldots,T}$.

Given the training set $X = \{X_1^T, \ldots, X_T^T\}^T$ and $\boldsymbol{y} = \{\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_T^T\}^T$, the posterior distribution at a test point $\boldsymbol{x}_*$ can be analytically derived as

$$\boldsymbol{f}(\boldsymbol{x}_*)|X, \boldsymbol{y}, \boldsymbol{x}_* \sim \mathcal{N}(\hat{\boldsymbol{f}}(\boldsymbol{x}_*), \Sigma_*) \tag{8}$$

Similar to the typical GP, the predicted mean $\hat{\boldsymbol{f}}(\boldsymbol{x}_*)$ and corresponding variance can be formulated as

$$\begin{aligned} \hat{\boldsymbol{f}}(\boldsymbol{x}_*) &= K_*^T[K_M(X, X) + \Sigma_M]^{-1}\boldsymbol{y} \\ \Sigma_* &= K_{**} - K_*^T[K_M(X, X) + \Sigma_M]^{-1}K_* \end{aligned} \tag{9}$$

where $K_* = K_M(X, \boldsymbol{x}_*)$ is an $nT \times T$ matrix representing the covariance between the training set and the test point, $K_{**} = K_M(\boldsymbol{x}_*, \boldsymbol{x}_*)$ represents the covariance at the test point with a size of $T \times T$, $\Sigma_M = \Sigma_s \otimes I_n \in R^{nT\times nT}$ is a diagonal matrix whose elements correspond to the noise, and the tth diagonal element of Σ_* corresponds to $\sigma_t^2(\boldsymbol{x}_*)$. Note that the predicted variance at $\boldsymbol{x}_*$ of $\boldsymbol{y}_t(\boldsymbol{x}_*)$ is $\sigma_t^2(\boldsymbol{x}_*) + \sigma_{s,t}^2$.

III. Gradient-Enhanced Multi-Output Gaussian Process Model

For convenience, the isotopic data ($X_1 = \cdots = X_T = \bar{X}$) are used for illustration although the proposed model can also be extended to the heterotopic case. For a prediction problem with m design variables and T outputs, if the training size for each output is $n_1 = \ldots = n_T = n$, that is, we have $N = nT$ samples in total for T outputs. Hence, there are $N \times m$ partial derivative values at these sample locations. The training data for the tth output should be

$$\begin{aligned} X_t &= \{X_t^{(1)}, \ldots, X_t^{(n)}, X_t^{(1)}, \ldots, X_t^{(n)}, \ldots, X_t^{(1)}, \ldots, X_t^{(n)}\}^T \\ &\quad \in R^{(n+mn)\times m} \\ \boldsymbol{y}_t &= \left\{\boldsymbol{y}_t^{(1)}, \ldots, \boldsymbol{y}_t^{(n)}, \frac{\partial \boldsymbol{y}_t^{(1)}}{\partial x_1}, \ldots, \frac{\partial \boldsymbol{y}_t^{(n)}}{\partial x_1}, \ldots, \frac{\partial \boldsymbol{y}_t^{(1)}}{\partial x_m}, \ldots, \frac{\partial \boldsymbol{y}_t^{(n)}}{\partial x_m}\right\} \\ &\quad \in R^{n+mn}, \qquad t = \{1, \ldots, T\} \end{aligned} \tag{10}$$

Then the definition of X and $\boldsymbol{y}$, which combine all the training data of the T outputs in Eqs. (1) and (2), should be changed to

$$\begin{aligned} \tilde{X} = \Big\{&X_1^{(1)}, \ldots, X_1^{(n)}, \ldots, X_T^{(1)}, \ldots, X_T^{(n)}, X_1^{(1)}, \ldots, X_1^{(n)}, \ldots, \\ &X_T^{(1)}, \ldots, X_T^{(n)}, \cdots, X_1^{(1)}, \ldots, X_1^{(n)}, \ldots, X_T^{(1)}, \ldots, \\ &X_T^{(n)}\Big\}^T \in R^{(N+mN)\times m} \end{aligned} \tag{11}$$

$$\begin{aligned} \tilde{\boldsymbol{y}} = \Big\{&\boldsymbol{y}_1^{(1)}, \ldots, \boldsymbol{y}_1^{(n)}, \ldots, \boldsymbol{y}_T^{(1)}, \ldots, \boldsymbol{y}_T^{(n)}, \frac{\partial \boldsymbol{y}_1^{(1)}}{\partial x_1}, \ldots, \frac{\partial \boldsymbol{y}_1^{(n)}}{\partial x_1}, \ldots, \\ &\frac{\partial \boldsymbol{y}_T^{(1)}}{\partial x_1}, \ldots, \frac{\partial \boldsymbol{y}_T^{(n)}}{\partial x_1}, \cdots, \frac{\partial \boldsymbol{y}_1^{(1)}}{\partial x_m}, \ldots, \frac{\partial \boldsymbol{y}_1^{(n)}}{\partial x_m}, \ldots, \frac{\partial \boldsymbol{y}_T^{(1)}}{\partial x_m}, \ldots, \frac{\partial \boldsymbol{y}_T^{(n)}}{\partial x_m}\Big\}^T \\ &\in R^{N+mN} \end{aligned} \tag{12}$$

where $(X_t^{(n)}, \boldsymbol{y}_t^{(n)})$ denotes the nth sample of the tth output, and the subscript of $[x_m, (\partial \boldsymbol{y}/\partial x_m)]$ represents the partial derivatives of the mth dimension. It can be seen that each partial derivative information is regarded as an additional sample point. If there is no derivative information added into the observations, the GEMOGP model would educe the MTGP model.

A. Inference

Assume that the output $\tilde{f} = \{\tilde{f}_1, \cdots \tilde{f}_T\}^T$ follows a Gaussian process with a prior of zero mean and a covariance of K_{GM}, where $K_{\mathrm{GM}} = k^t \otimes K^x$ consists of four parts and each part can be expressed as [23]

$$\begin{aligned} &\mathrm{Cov}[\tilde{f}_s(\boldsymbol{x}^{(i)}), \tilde{f}_t(\boldsymbol{x}^{(j)})] = k^t(s,t)K^x(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}), \\ &\mathrm{Cov}\left[\tilde{f}_s(\boldsymbol{x}^{(i)}), \frac{\partial \tilde{f}_t(\boldsymbol{x}^{(j)})}{\partial x_u}\right] = k^t(s,t)\frac{\partial K^x(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})}{\partial x_u^{(j)}}, \\ &\mathrm{Cov}\left[\frac{\partial \tilde{f}_s(\boldsymbol{x}^{(i)})}{\partial x_u}, \tilde{f}_t(\boldsymbol{x}^{(j)})\right] = k^t(s,t)\frac{\partial K^x(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})}{\partial x_u^{(j)}}, \\ &\mathrm{Cov}\left[\frac{\partial \tilde{f}_s(\boldsymbol{x}^{(i)})}{\partial x_u}, \frac{\partial \tilde{f}_t(\boldsymbol{x}^{(j)})}{\partial x_v}\right] = k^t(s,t)\frac{\partial^2 K^x(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})}{\partial x_u^{(i)}\partial x_v^{(j)}} \end{aligned} \tag{13}$$

where $\mathrm{Cov}[\tilde{f}_s(\boldsymbol{x}^{(i)}), \tilde{f}_t(\boldsymbol{x}^{(j)})]$ denotes the cross-covariance between the outputs s and t at $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$.

The covariance matrix K_{GM} is represented as the Kronecker product of k^t and K^x. Here, $k^t \in R^{T\times T}$ is a matrix that represents the correlation or similarity across outputs. It is a challenge to construct a valid positive semidefinite covariance matrix k^t in the MTGP structure. One strategy is to use the "free-form" parameterization proposed by Bonilla et al. [13], which uses Cholesky decomposition to parameterize k^t as $k^t = LL^T$, with

$$L = \begin{pmatrix} a_1 & 0 & \cdots & 0 \\ a_2 & a_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{w-T+1} & a_{w-T+2} & \cdots & a_w \end{pmatrix} \tag{14}$$

where $w = T(T+1)/2$ is the number of parameters in k^t. One advantage of this free-form parameterization is that the elements of k^t can be scaled freely [7], and it is thus adopted in the proposed model. There are also some other ways to construct k^t; e.g., Cohn and Specia [35] proposed a so-called "combined" kernel, which can be parameterized as $k^t = \mathbf{1} + \alpha I$, where $\mathbf{1}$ is a matrix with every entry equals to 1; Osborne et al. [36] parameterized k^t as $k^t = \mathrm{diag}(\boldsymbol{e})S^T S\mathrm{diag}(\boldsymbol{e})$, where $\boldsymbol{e}$ is a vector of the length scales for each dimension and S is an upper triangular matrix describing the particular spherical coordinates of points.

As for the matrix K^x, it is a typical stationary covariance matrix over inputs. Because the process variance can be included in k^t, let K^x be a spatial correlation function with unit variance. $\partial \tilde{f}_s(\boldsymbol{x}^{(i)})/\partial x_k$ and $\partial \tilde{f}_t(\boldsymbol{x}^{(j)})/\partial x_k$ are the partial derivatives with respect to the kth component of the sth output at $\boldsymbol{x}^{(i)}$ and the tth output at $\boldsymbol{x}^{(j)}$, respectively. Because the expressions of the covariance matrix are hard to write down in high-dimensional problems, the one-dimensional (1-D) problem is demonstrated here for convenience. The covariance matrix $K^x \in R^{(N+mN)\times(N+mN)}$ that consists of correlation between responses and responses, responses and gradients, gradients and responses, and gradients and gradients, can be formulated as [37]

$$K^x = \begin{pmatrix} K^x(\boldsymbol{x}_{1,1}, \boldsymbol{x}_{1,1}) & \cdots & K^x(\boldsymbol{x}_{1,1}, \boldsymbol{x}_{T,n}) & \frac{\partial K^x(\boldsymbol{x}_{1,1}, \boldsymbol{x}_{1,1})}{\partial \boldsymbol{x}_{1,1}} & \cdots & \frac{\partial K^x(\boldsymbol{x}_{1,1}, \boldsymbol{x}_{T,n})}{\partial \boldsymbol{x}_{T,n}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ K^x(\boldsymbol{x}_{T,n}, \boldsymbol{x}_{1,1}) & \cdots & K^x(\boldsymbol{x}_{T,n}, \boldsymbol{x}_{T,n}) & \frac{\partial K^x(\boldsymbol{x}_{T,n}, \boldsymbol{x}_{1,1})}{\partial \boldsymbol{x}_{1,1}} & \cdots & \frac{\partial K^x(\boldsymbol{x}_{T,n}, \boldsymbol{x}_{T,n})}{\partial \boldsymbol{x}_{T,n}} \\ \frac{\partial K^x(\boldsymbol{x}_{1,1}, \boldsymbol{x}_{1,1})}{\partial \boldsymbol{x}_{1,1}} & \cdots & \frac{\partial K^x(\boldsymbol{x}_{1,1}, \boldsymbol{x}_{T,n})}{\partial \boldsymbol{x}_{1,1}} & \frac{\partial^2 K^x(\boldsymbol{x}_{1,1}, \boldsymbol{x}_{1,1})}{\partial \boldsymbol{x}_{1,1}^2} & \cdots & \frac{\partial^2 K^x(\boldsymbol{x}_{1,1}, \boldsymbol{x}_{T,n})}{\partial \boldsymbol{x}_{1,1}\partial \boldsymbol{x}_{T,n}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial K^x(\boldsymbol{x}_{T,n}, \boldsymbol{x}_{1,1})}{\partial \boldsymbol{x}_{T,n}} & \cdots & \frac{\partial K^x(\boldsymbol{x}_{T,n}, \boldsymbol{x}_{T,n})}{\partial \boldsymbol{x}_{T,n}} & \frac{\partial^2 K^x(\boldsymbol{x}_{T,n}, \boldsymbol{x}_{1,1})}{\partial \boldsymbol{x}_{T,n}\partial \boldsymbol{x}_{1,1}} & \cdots & \frac{\partial^2 K^x(\boldsymbol{x}_{T,n}, \boldsymbol{x}_{T,n})}{\partial \boldsymbol{x}_{T,n}^2} \end{pmatrix} \tag{15}$$

The complete covariance matrix for the training data incorporating the gradient information $K_{\mathrm{GM}}(\tilde{X}, \tilde{X}) = k^t \otimes K^x \in R^{(N+mN)\times(N+mN)}$ consists of the four parts in Eq. (13):

$$K_{\mathrm{GM}} = \begin{pmatrix} k^t(1,1)K^x(\boldsymbol{x}_{1,1},\boldsymbol{x}_{1,1}) & \cdots & k^t(1,T)K^x(\boldsymbol{x}_{1,1},\boldsymbol{x}_{T,n}) & k^t(1,1)\frac{\partial K^x(\boldsymbol{x}_{1,1},\boldsymbol{x}_{1,1})}{\partial \boldsymbol{x}_{1,1}} & \cdots & k^t(1,T)\frac{\partial K^x(\boldsymbol{x}_{1,1},\boldsymbol{x}_{T,n})}{\partial \boldsymbol{x}_{T,n}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ k^t(T,1)K^x(\boldsymbol{x}_{T,n},\boldsymbol{x}_{1,1}) & \cdots & k^t(T,T)K^x(\boldsymbol{x}_{T,n},\boldsymbol{x}_{T,n}) & k^t(T,1)\frac{\partial K^x(\boldsymbol{x}_{T,n},\boldsymbol{x}_{1,1})}{\partial \boldsymbol{x}_{1,1}} & \cdots & k^t(T,T)\frac{\partial K^x(\boldsymbol{x}_{T,n},\boldsymbol{x}_{T,n})}{\partial \boldsymbol{x}_{T,n}} \\ k^t(1,1)\frac{\partial K^x(\boldsymbol{x}_{1,1},\boldsymbol{x}_{1,1})}{\partial \boldsymbol{x}_{1,1}} & \cdots & k^t(1,T)\frac{\partial K^x(\boldsymbol{x}_{1,1},\boldsymbol{x}_{T,n})}{\partial \boldsymbol{x}_{1,1}} & k^t(1,1)\frac{\partial^2 K^x(\boldsymbol{x}_{1,1},\boldsymbol{x}_{1,1})}{\partial x_{1,1}^2} & \cdots & k^t(1,T)\frac{\partial^2 K^x(\boldsymbol{x}_{1,1},\boldsymbol{x}_{T,n})}{\partial x_{1,1}\partial x_{T,n}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ k^t(T,1)\frac{\partial K^x(\boldsymbol{x}_{T,n},\boldsymbol{x}_{1,1})}{\partial \boldsymbol{x}_{T,n}} & \cdots & k^t(T,T)\frac{\partial K^x(\boldsymbol{x}_{T,n},\boldsymbol{x}_{T,n})}{\partial \boldsymbol{x}_{T,n}} & k^t(T,1)\frac{\partial^2 K^x(\boldsymbol{x}_{T,n},\boldsymbol{x}_{1,1})}{\partial x_{T,n}\partial x_{1,1}} & \cdots & k^t(T,T)\frac{\partial^2 K^x(\boldsymbol{x}_{T,n},\boldsymbol{x}_{T,n})}{\partial x_{T,n}^2} \end{pmatrix} \tag{16}$$

Because a Gaussian process is a collection of random variables, wherein any finite number of which have a joint Gaussian distribution, the joint distribution of the observations and the function values at the test locations is [4]

$$\begin{bmatrix} \tilde{\boldsymbol{y}} \\ \tilde{\boldsymbol{f}}_* \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} K_{\mathrm{GM}} + \tilde{\Sigma}_M & \tilde{K}_* \\ \tilde{K}_* & \tilde{K}_{**} \end{bmatrix}\right) \tag{17}$$

where $\tilde{K}_* = K_{\mathrm{GM}}(\tilde{X}, \boldsymbol{x}_*) \in R^{(N+mN)\times T}$ is a covariance matrix between the training set and the test set, $\tilde{K}_{**} = K_{\mathrm{GM}}(\boldsymbol{x}_*, \boldsymbol{x}_*) \in R^{T\times T}$ has elements $k^t(t,t') \cdot K^x(\boldsymbol{x}_*, \boldsymbol{x}_*)$ for $t, t' = 1, \ldots, T$, the tth diagonal element of $\tilde{\Sigma}_*$ corresponds to $\sigma_t^2(\boldsymbol{x}_*)$, and $\tilde{\Sigma}_M$ is a diagonal noise matrix. For the 1-D case, it can be expressed as [26]

$$\tilde{\Sigma}_M = \begin{pmatrix} \upsilon_{1,1}^2 & & & & & & & \\ & \upsilon_{1,2}^2 & & & & 0 & & \\ & & \ddots & & & & & \\ & & & \upsilon_{T,n}^2 & & & & \\ & & & & \delta_{1,1}^2 & & & \\ & & & & & \delta_{1,2}^2 & & \\ & 0 & & & & & \ddots & \\ & & & & & & & \delta_{T,n}^2 \end{pmatrix} \in R^{(N+mN)\times(N+mN)} \tag{18}$$

where υ^2 and δ^2 are the observation errors and gradient errors, respectively. These errors are assumed to be independent and identically distributed Gaussian noise and are thus directly added to the diagonal of the covariance matrix. It has been pointed out that the consideration of these errors can not only help transfer information across multiple outputs [17], but also improve the robustness of the model [26].

By conditioning the joint Gaussian prior distribution on $\tilde{\boldsymbol{y}}$, the posterior distribution of $\tilde{\boldsymbol{f}}_*$ can be analytically derived as

$$\tilde{\boldsymbol{f}}_* | \tilde{X}, \tilde{\boldsymbol{y}}, \boldsymbol{x}_* \sim \mathcal{N}\left(\hat{\tilde{\boldsymbol{f}}}_*, \tilde{\Sigma}_*\right) \tag{19}$$

where the predicted mean and variance at an unseen point $\boldsymbol{x}_*$ can be respectively expressed as

$$\begin{aligned} \hat{\tilde{\boldsymbol{f}}}_* &= \tilde{K}_*^T [K_{\mathrm{GM}}(\tilde{X}, \tilde{X}) + \tilde{\Sigma}_M]^{-1} \tilde{\boldsymbol{y}} \\ \tilde{\Sigma}_* &= \tilde{K}_{**} - \tilde{K}_*^T [K_{\mathrm{GM}}(\tilde{X}, \tilde{X}) + \tilde{\Sigma}_M]^{-1} \tilde{K}_* \end{aligned} \tag{20}$$

B. Kernel Functions

The calculation of the covariance matrix K^x requires the covariance function as well as its first and second derivatives with respect to $\boldsymbol{x}$. The typical stationary covariance functions can be chosen here. Take the well-known Gaussian covariance function [4], for example,

$$k^x(x, x') = \exp\left(-\frac{1}{2}(x - x')^T P (x - x')\right) \tag{21}$$

where the diagonal matrix $P = \mathrm{diag}(l_1^2, \ldots, l_m^2)$ with the parameters $l_i^2 (i = 1, \ldots, m)$ defining the characteristic length scales. Its first- and second-order partial derivatives can be formulated as

$$\frac{\partial k^x(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})}{\partial \boldsymbol{x}_u^{(j)}} = \frac{1}{l_u^2}(\boldsymbol{x}_u^{(i)} - \boldsymbol{x}_u^{(j)}) k^x(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) \tag{22}$$

$$\frac{\partial^2 k^x(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})}{\partial \boldsymbol{x}_u^{(i)} \partial \boldsymbol{x}_v^{(j)}} = \begin{cases} -\frac{1}{l_u^2}\frac{1}{l_v^2}(\boldsymbol{x}_u^{(i)} - \boldsymbol{x}_u^{(j)})(\boldsymbol{x}_v^{(i)} - \boldsymbol{x}_v^{(j)}) k^x(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) & u \neq v \\ \left[\frac{1}{l^2} - \frac{1}{l^4}(\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)})^2\right] k^x(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) & u = v \end{cases} \tag{23}$$

C. Learning Hyperparameters

Given the set of training data, several parameters should be inferred in order to implement the proposed model, namely, a_t of k^t, l^2 of k^x, and υ^2 and δ^2 of $\tilde{\Sigma}_M$. Similar to the typical SOGP, a commonly used strategy is to exploit the maximum likelihood estimation (MLE) to maximize the marginal likelihood $p(\tilde{\boldsymbol{y}}|\tilde{X}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represents all the hyperparameters. This problem is equivalent to minimize the negative log marginal likelihood (NLML) as [17]

$$\begin{aligned} \mathrm{NLML} &= -\log p(\tilde{\boldsymbol{y}}|\tilde{X}, \boldsymbol{\theta}) \\ &= \frac{1}{2}\tilde{\boldsymbol{y}}^T [K_{\mathrm{GM}}(\tilde{X}, \tilde{X}) + \tilde{\Sigma}_M]^{-1} \tilde{\boldsymbol{y}} + \frac{1}{2}\log |K_{\mathrm{GM}}(\tilde{X}, \tilde{X}) \\ &\quad + \tilde{\Sigma}_M| + \frac{N}{2}\log 2\pi \end{aligned} \tag{24}$$

To minimize Eq. (24), some efficient gradient-based methods, such as the conjugate gradient method, can be used [13,38]. However, it is often a nontrivial problem to optimize marginal likelihood with respect to the hyperparameters because of two difficulties [39]: a) multimodality of the log-likelihood function, and b) long ridges in the log-likelihood function. These issues can be addressed by adopting a global stochastic optimization algorithm. In this work, an effective differential evolution (DE) algorithm, JADE [40], is adopted to solve this problem because it is prone to get the global optimum and has stable and high performance.

IV. Cases Studies

In this section, three examples with diverse characteristics and data structures are used to test the performance of the proposed GEMOGP model. The three examples include two analytical cases (a 1-D pedagogical example and a 10-dimensional [10-D] Dixon–Price example) and a prediction problem of aerodynamic coefficients of

a NACA 0012 airfoil. In the two synthetic examples, the observation errors of the responses $\varepsilon \sim GP(0, 0.01^2)$, and the gradient errors are set to $\delta \sim GP(0, 0.1^2)$, which are typically larger than errors in observations [26]. For the purpose of comparison, three other GP-based modeling approaches are also tested: 1) the typical SOGP modeling approach [4], 2) the direct GEK modeling approach put forth by Morris et al. [23], and 3) the MTGP modeling approach [13] introduced in Sec. II. All the codes are implemented in the MATLAB environment at a computer with Intel Core i9 CPU (3.6 GHz). To estimate the hyperparameters, we use JADE to solve the min-NLML problem for all the four models. The population size is set to 100, and the maximum number of function evaluations is $800n_p$, where n_p is the total number of hyperparameters in each model. The search ranges in hyperparameter tuning depend on different types of hyperparameters. For the hyperparameters a_t in k^t, because the process variance can be included in k^t, the search range is set properly larger than the sample variance. As for the hyperparameters l_i^2 in k^x, the search range is set to [exp (−5), exp (8)]. For v^2 and δ^2, the search range is set to [exp (−8), 0.01] and [exp (−8), 0.1], respectively.

The sample locations in the two synthetic experiments are generated by MATLAB routine *lhsdesign* with "*maximin*" criterion, and the gradients are calculated using the complex step method [41]. Three error metrics, coefficient of determination R^2, root-mean-square error (RMSE), and maximum absolute error (MAE), are adopted to assess the prediction accuracy of different modeling approaches. R^2 and RMSE can reflect the global accuracy of the model, whereas MAE is an indicator of the local accuracy of the model. A larger value of R^2 means a better fitting to the model. A lower RMSE or MAE value indicates a higher prediction accuracy. These three error metrics can be calculated as [42]

$$R^2 = 1 - \sum_{1}^{N_{\text{test}}} (y_i - \hat{y}_i)^2 \Big/ \sum_{1}^{N_{\text{test}}} (y_i - \bar{y})^2 \tag{25}$$

$$\text{RMSE} = \sqrt{\frac{1}{N_{\text{test}}} \sum_{1}^{N_{\text{test}}} (y_i - \hat{y}_i)^2} \tag{26}$$

$$\text{MAE} = \max |y_i - \hat{y}_i| \tag{27}$$

where N_{test} is the number of test points, y_i is the true function value at the ith test point, $\hat{y}_i$ is the predicted value of the ith test point, and $\bar{y}$ is the mean of the true function values at all test points.

A. One-Dimensional Example

First, a 1-D example with three outputs [23,43] is used to demonstrate the performance of the proposed model. The three outputs are expressed as

$$\begin{aligned}
&y_1(x) = f_1(x) + \varepsilon_1, \qquad y_2(x) = f_2(x) + \varepsilon_2,\\
&y_3(x) = f_3(x) + \varepsilon_3\\
&f_1(x) = 5x^2 \sin(12x),\\
&f_2(x) = 5x^2 \sin(12x) + (x^3 - 0.5)\sin(3x - 0.5) + 4\cos(2x),\\
&f_3(x) = (6x - 2)^2 \sin(12x - 4), \qquad x \in [0, 1]
\end{aligned} \tag{28}$$

The 1-D example consists of three outputs with different correlations. The Pearson correlation coefficient r is used to measure the correlations, which can be calculated as

$$r = \frac{\text{cov}(Z_1, Z_2)}{\sqrt{\text{var}(Z_1)\text{var}(Z_2)}} \tag{29}$$

where Z_1 and Z_2 are two random variables. The first two functions are of higher correlation with a Pearson correlation coefficient $R_{1,2} = 0.84$, whereas they are less correlated with the third function with $R_{1,3} = -0.75$ and $R_{2,3} = -0.61$.

The 1-D example is tested with three different data structures: isotopic data, heterotopic data, and partial interval data for $y_1(x)$. Four sample points are generated for each output in the first two cases and five in the third case. Additional 100 test points evenly distributed in the interval [0,1] are adopted to calculate the three error metrics.

1. *Isotopic Data*

In this subsection, the four modeling approaches, SOGP, GEK, MTGP, and GEMOGP, are tested using isotopic data, where the three outputs are sampled at the same locations. The four sampling locations are 0.0517, 0.3002, 0.5679, and 0.9780, respectively. The selected hyperparameters after tuning are shown in Table A1 in the Supplemental Materials. The predicted results of the three outputs are shown in Figs. 1–3, respectively. The hollow circles represent the training points, the red solid line represents the predicted values (with the gray shadow representing the 95% confidence interval of the predictions), and the green dotted line represents the true function values. The results of the three error metrics for the four modeling approaches are summarized in Table 1, and the best results of the three error metrics among these four modeling approaches are marked in bold.

As shown in Figs. 1–3, the SOGP model performs the worst due to the lack of enough sample information. Compared with SOGP, the MTGP model provides better predictions. This is because the multi-output models are capable of transferring useful information across outputs. The two gradient-enhanced models, GEMOGP and GEK, perform better than the other two models due to the incorporation of the additional gradient information at the sample locations. Compared with the GEK model, the proposed GEMOGP model provides a better prediction with a reduction of 16.1% on the RMSE value for $y_1(x)$, 27.8% for $y_2(x)$, and a slightly better prediction for $y_3(x)$. This is because the proposed model can not only use the gradient information to enhance the performance of the model but also transfer useful knowledge across outputs. It can also be concluded from Table 1 that the proposed model performs best among these four models.

One can observe that the GEMOGP model performs better than the single-output GEK model in terms of the predictions for the first two outputs, while provides comparable predictions for $y_3(x)$. This is related to the correlation across outputs. The third output is less correlated with $y_1(x)$ and $y_2(x)$ with the Pearson correlation coefficient $r < 0.8$, whereas the first two outputs have a higher correlation with $R_{1,2} = 0.84$. As a result, useful information can be transferred between $y_1(x)$ and $y_2(x)$, whereas modeling $y_3(x)$ jointly gains little benefit. It can be concluded that the effect of a multi-output model usually improves with a higher Pearson correlation coefficient between two outputs.

2. *Heterotopic Data*

Figures 4–6 show the predictions of the three outputs using heterotopic data, respectively. The sample locations of $y_1(x)$ are the same as the isotopic case, whereas the sample locations of $y_2(x)$ and $y_3(x)$ are different from those of the isotopic case, which are 0.0825, 0.3552, 0.5890, 0.8344, and 0.0874, 0.4444, 0.5808, 0.7626, respectively. The selected hyperparameters after tuning are shown in Table A2 in the Supplemental Materials.

As observed in Figs. 4–6, the proposed GEMOGP approach still outperforms the other three approaches. Comparing with the isotopic case, it can be seen that the predictions of $y_1(x)$ from the SOGP and GEK models change very little due to the same training points, whereas the predictions from the GEMOGP model have a significant improvement. This is because the two single-output models cannot transfer information from $y_2(x)$ and $y_3(x)$. This is expected because modeling the outputs separately leads to similar predictions with the same training points, whereas for the GEMOGP model the predictions of $y_1(x)$ improve significantly in spite of having the same training data. This is owing to the exploitation of additional gradient information resulting in good predictions for $y_2(x)$ and

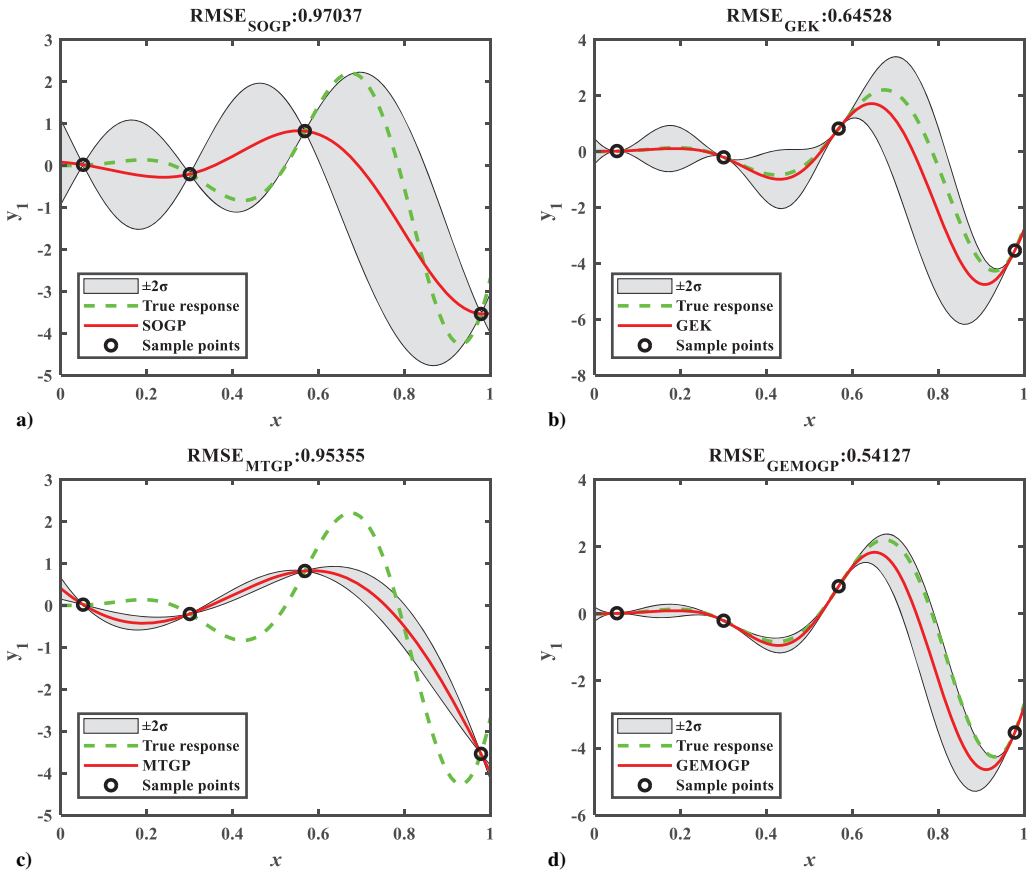


Fig. 1 Predictions of y_1 on the 1-D example for the four different methods: a) SOGP, b) GEK, c) MTGP, and d) GEMOGP.

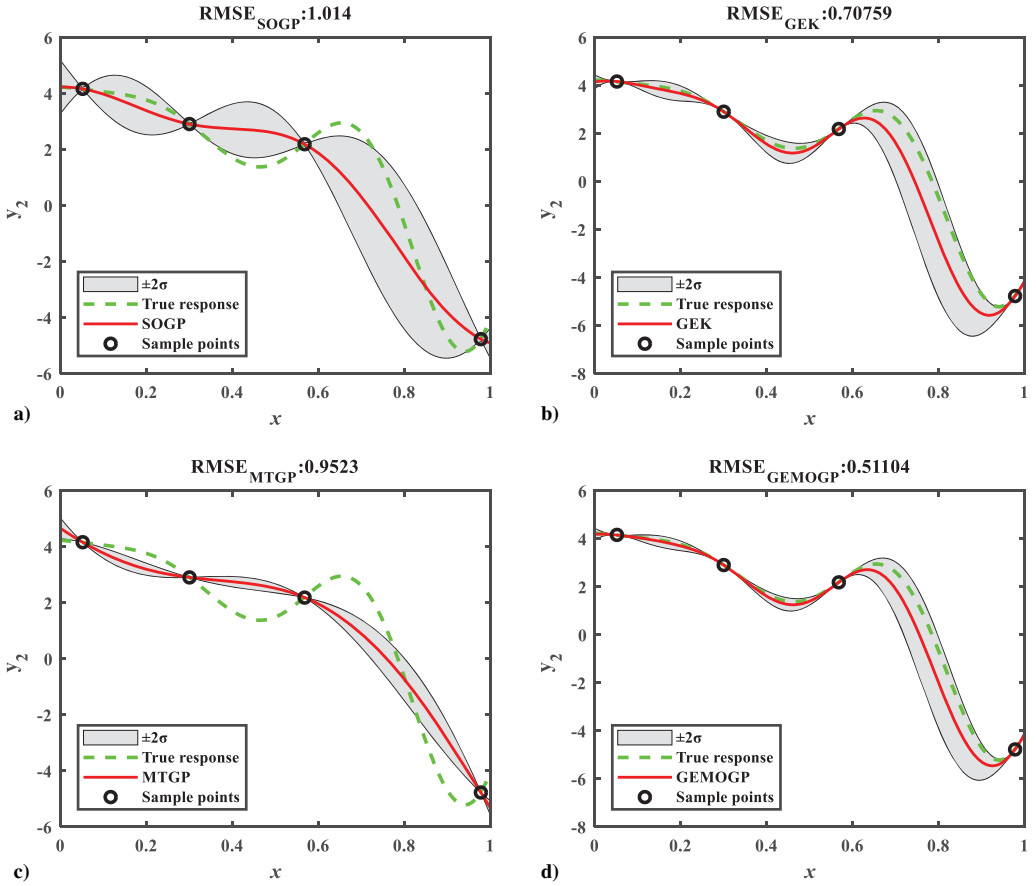


Fig. 2 Predictions of y_2 on the 1-D example for the four different methods: a) SOGP, b) GEK, c) MTGP, and d) GEMOGP.

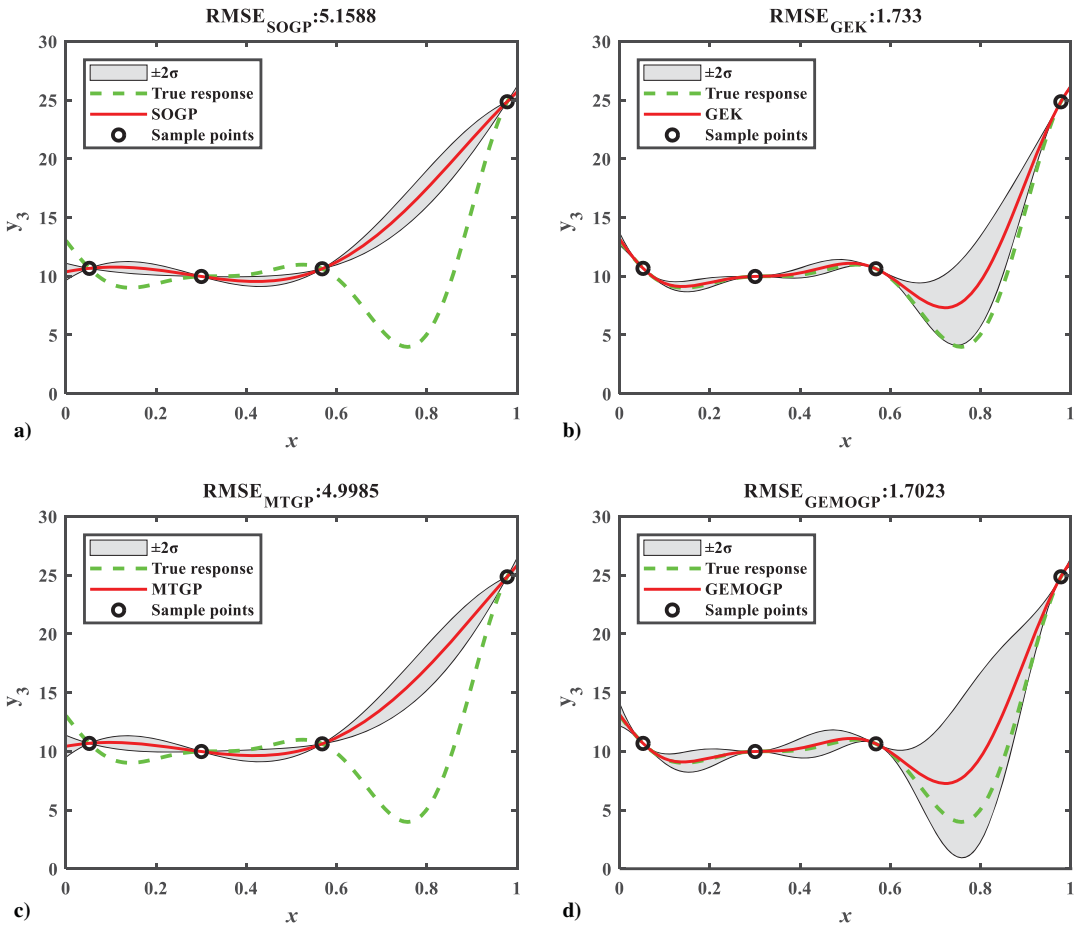


Fig. 3 Predictions of y_3 on the 1-D example for the four different methods: a) SOGP, b) GEK, c) MTGP, and d) GEMOGP.

Table 1 Comparison results for the one-dimensional example under isotopic data

	SOGP			GEK		
Metrics	y_1	y_2	y_3	y_1	y_2	y_3
R^2	0.6602	0.8763	−0.2750	0.8497	0.9397	0.8561
RMSE	0.9704	1.0140	5.1588	0.6453	0.7076	1.7330
MAE	2.2000	2.3273	12.4703	1.6297	1.8304	4.5539
	MTGP			GEMOGP		
Metrics	y_1	y_2	y_3	y_1	y_2	y_3
R^2	0.6719	0.8909	−0.1970	0.8943	0.9686	0.8612
RMSE	0.9536	0.9523	4.9985	0.5413	0.5110	1.7023
MAE	1.9258	1.8191	12.1189	1.4141	1.3281	4.4782

useful transferred information in the multi-output modeling. However, it should be noticed that the predictions of $y_1(x)$ in the MTGP model also improves very little. This is because the predictions of the other two outputs are not good enough so that the MTGP model cannot transfer the right information across outputs. As a result, the MTGP model is hard to capture the primary features of the outputs using a small number of sample points. Modeling these outputs jointly has little effect on the model accuracy and even performs a little worse than the SOGP model. This situation is called "information sparsity" in the paper [17].

The three error metrics of the four modeling approaches are listed in Table 2. It can be seen that the SOGP and MTGP models provide poor predictions for the three outputs with $R^2 < 0.8$. Both the two gradient-enhanced models, GEMOGP and GEK, provide good predictions for the first two outputs with $R^2 > 0.85$, whereas the proposed model performs even better with $R^2 > 0.98$ for both outputs. In terms of the predictions for $y_3(x)$, the four approaches perform badly in [0.8, 1.0] because it is an extrapolation in this region. Although the multi-output models are able to share correlation information across outputs, $y_3(x)$ has a low correlation with the other two outputs, which results in little benefit for the predictions of $y_3(x)$. This result also coincides with the conclusion in the isotopic case on the influence of the Pearson correlation coefficient. It can also be concluded from Table 2 that the proposed model performs best considering both global accuracy and local accuracy.

From this case, it is advised to use the heterotopic data structure in the multi-output scenario if these outputs can be simulated separately. Because the augment of information diversity can help to infer the hyperparameter better so that the prediction accuracy of the surrogate model can be improved. Besides, for the multi-output modeling, it is recommended to avoid the information sparsity and modeling multiple outputs with low correlations to obtain better performance.

3. *Partial Interval Data for $y_1(x)$*

In this subsection, the case where $y_1(x)$ is sampled only in a subinterval is discussed. For illustration, only the first two outputs $y_1(x)$ and $y_2(x)$ are used to construct the surrogate models. The sampling locations of $y_1(x)$ are generated in subinterval [0, 0.5], whereas the sampling interval of $y_2(x)$ is for a full range [0,1]. The sample locations of $y_1(x)$ are 0.0286, 0.1360, 0.2808, 0.3427, 0.4467, and of $y_2(x)$ are 0.0741, 0.2411, 0.4212, 0.6425, 0.9866. The selected hyperparameters after tuning are shown in Table A3 in the Supplemental Materials. The predictions of $y_1(x)$ and $y_2(x)$ are shown in Figs. 7 and 8, respectively. It can be observed that compared with the first two cases, the four models provide better predictions of $y_2(x)$ as the number of sample points increases to five. Because there is no sample point in the interval [0.5, 1] for $y_1(x)$, it will be difficult to approximate the $y_1(x)$ in this region. As shown in Fig. 7, the two single-output models provide poor predictions for $y_1(x)$ in the interval

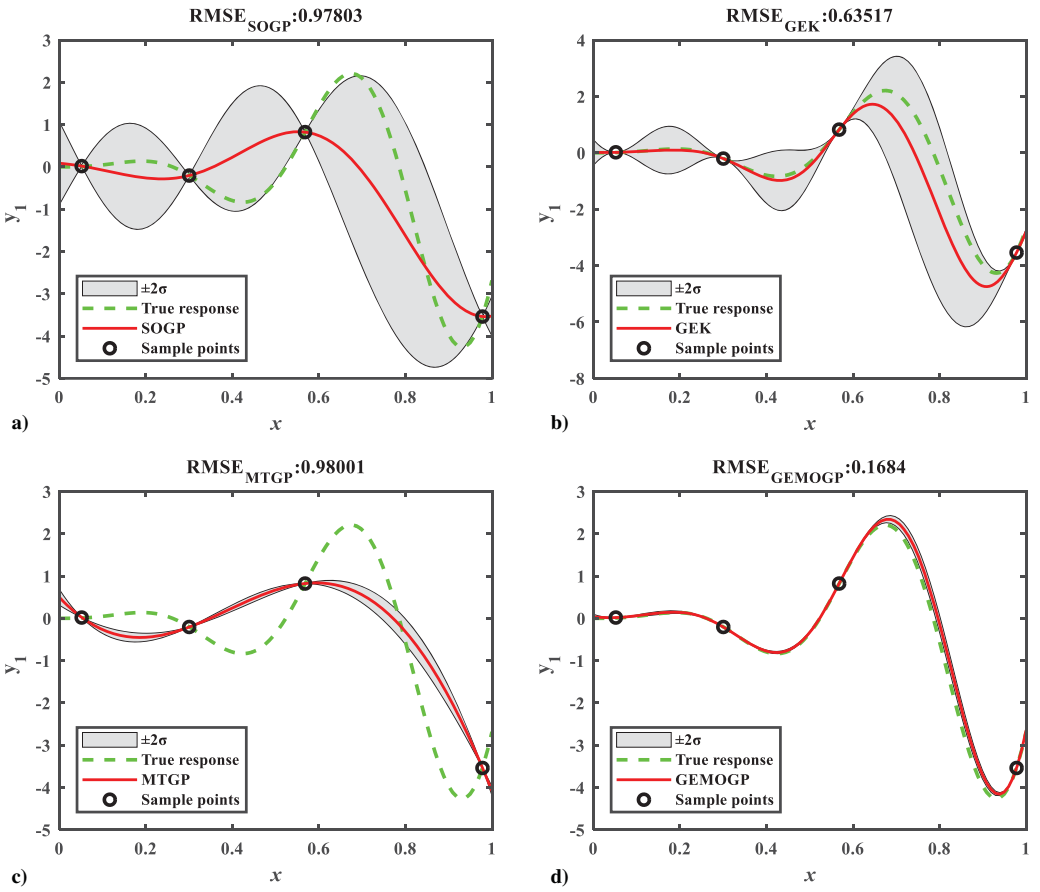


Fig. 4 Predictions of y_1 on the 1-D example for the four different methods: a) SOGP, b) GEK, c) MTGP, and d) GEMOGP.

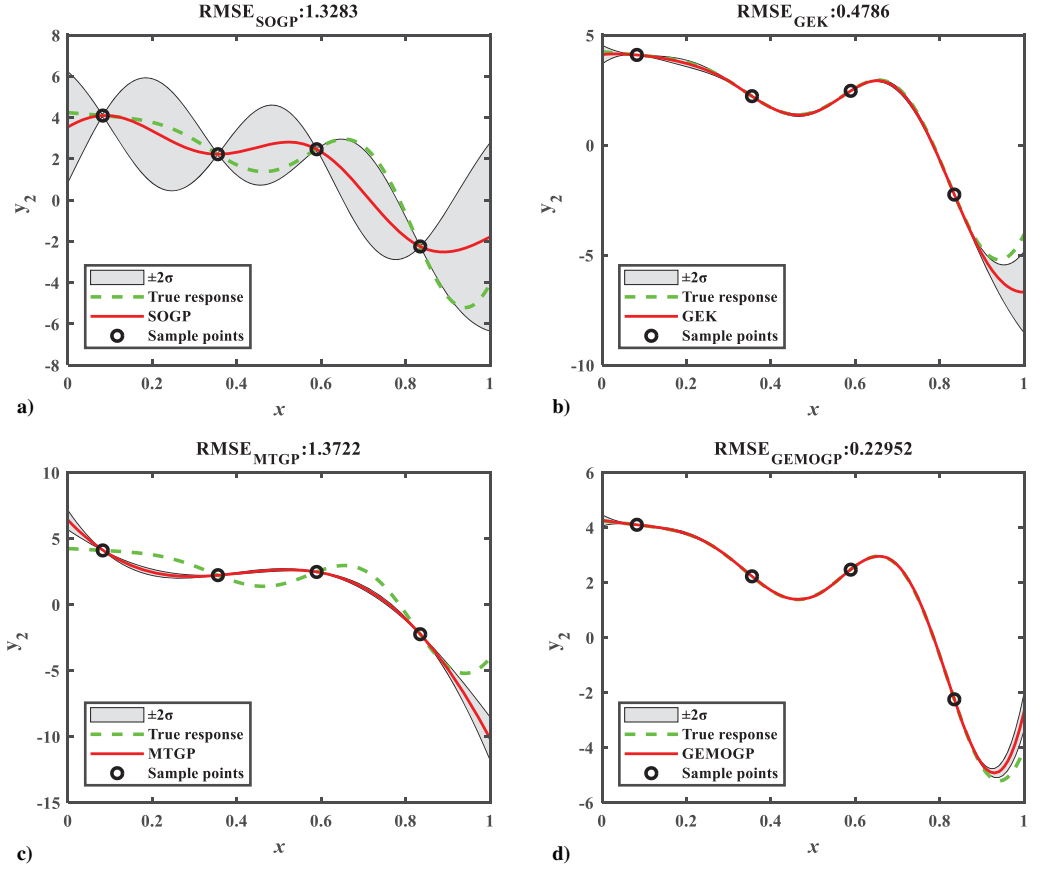


Fig. 5 Predictions of y_2 on the 1-D example for the four different methods: a) SOGP, b) GEK, c) MTGP, and d) GEMOGP.

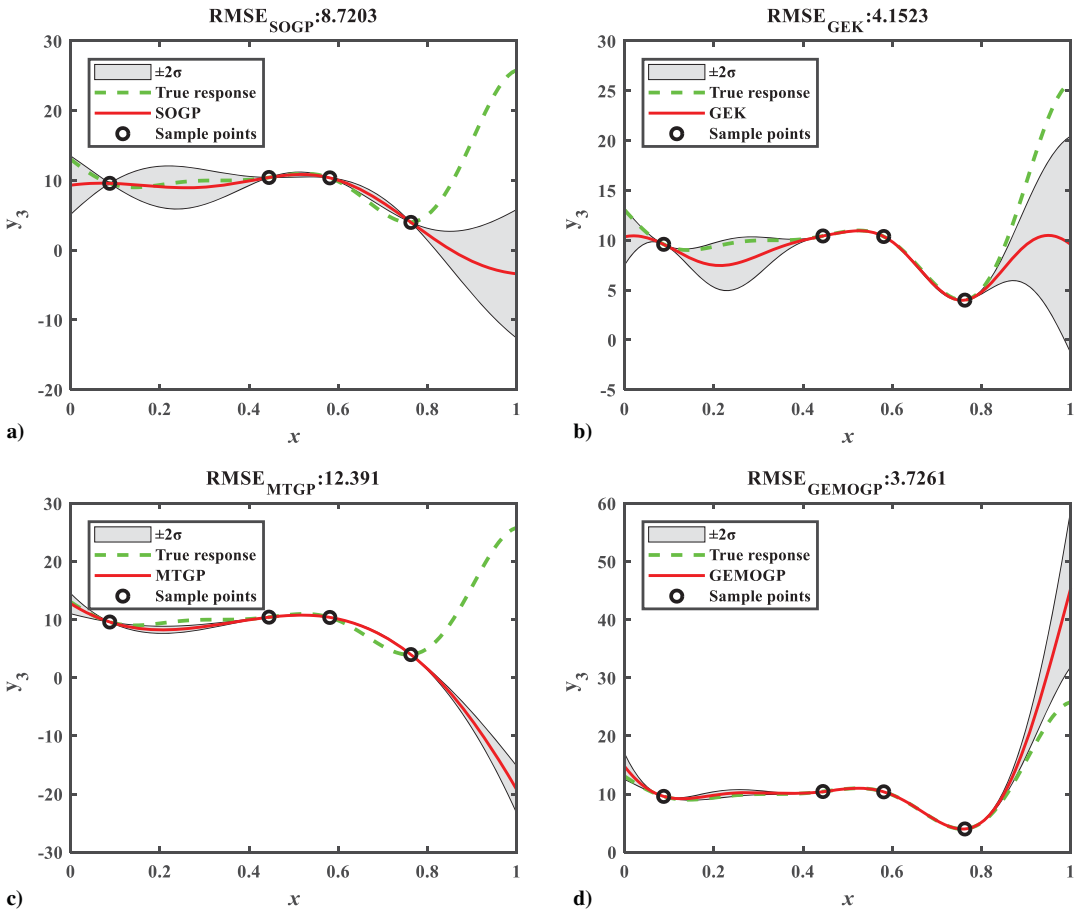


Fig. 6 Predictions of y_3 on the 1-D example for the four different methods: a) SOGP, b) GEK, c) MTGP, and d) GEMOGP.

Table 2 Comparison results for the one-dimensional example under heterotopic data

	SOGP			GEK		
Metrics	y_1	y_2	y_3	y_1	y_2	y_3
R^2	0.6548	0.7877	−2.6431	0.8544	0.9724	0.1740
RMSE	0.9780	1.3283	8.7203	0.6352	0.4786	4.1523
MAE	2.2186	2.9311	29.2162	1.6026	2.6363	16.2508
	MTGP			GEMOGP		
Metrics	y_1	y_2	y_3	y_1	y_2	y_3
R^2	0.6534	0.7734	−6.3556	0.9898	0.9937	0.3349
RMSE	0.9800	1.3722	12.3910	0.1684	0.2295	3.7261
MAE	2.1121	6.1220	45.0991	0.4375	1.3795	19.5520

[0.5, 1]. This is expected as no sample information can be obtained from the training data in this region. Although the MTGP model could learn some information from $y_2(x)$, it still performs poorly in predicting $y_1(x)$ in the interval [0.5, 1]. This may be because the information in [0.5, 1] transferred from $y_2(x)$ is not enough to construct an accurate model. However, because the GEMOGP model incorporates not only the correlation of the different responses but also the correlation between gradients and responses, it can transfer more information across outputs. As a result, the GEMOGP model learns a similar feature of the two outputs in [0.5, 1] and provides good predictions for $y_1(x)$ in this region. Comparative results of the three error metrics for the four modeling approaches are listed in Table 3. It can be seen that the four models provide acceptable predictions for $y_2(x)$ with $R^2 > 0.95$, but perform terribly in predictions for $y_1(x)$ except the proposed GEMOGP model. That is to say, the GEMOGP model provides the best predictions among these four models for both outputs.

B. Ten-Dimensional Dixon-Price Example

In this section, a 10-D Dixon–Price example [44] with two outputs is discussed. The expressions of the outputs are given as

$$y_1(\boldsymbol{x}) = f_1(\boldsymbol{x}) + \varepsilon_1, \qquad y_2(\boldsymbol{x}) = f_2(\boldsymbol{x}) + \varepsilon_2$$
$$f_1(\boldsymbol{x}) = (x_1 - 1)^2 + \sum_{i=2}^{10} i(2x_i^2 - x_{i-1})^2,$$
$$f_2(\boldsymbol{x}) = 0.9 f_1(x) - \sum_{i=1}^{9} 0.2 x_i x_{i+1} + 100,$$
$$x_i \in [-10, 10], i = 1, 2, \ldots, 10 \tag{30}$$

For comparison, the four modeling approaches are tested for the case, where the sample sizes of the two outputs are different from each other, that is, $n_1 \neq n_2$. Besides, the influence of sample size on the performance of the surrogate models is also discussed. For illustration, the heterotopic data are used to construct the surrogate models. To eliminate the deviation from the design of experiments, 10 independent repetitions are randomly generated for each sample size. The three error metrics are calculated using 10,000 test points generated by *lhsdesign*.

1. $n_1 \neq n_2$

In this part, the sample size is set to $n_1 = 60$ and $n_2 = 100$. Figure 9 depicts the boxplots of the log-RMSE values of the two outputs for the four models over 10 independent runs. The plots of the other two error metrics are shown in Figs. B1 and B2 in the Supplemental Materials. The top line and the bottom line of the box represent the third quartile and first quartile values, respectively. The red line inside the box represents the middle value of the dataset, and the square represents the average value. The two horizontal lines outside the box represent the smallest and largest data points excluding any outliers, respectively.

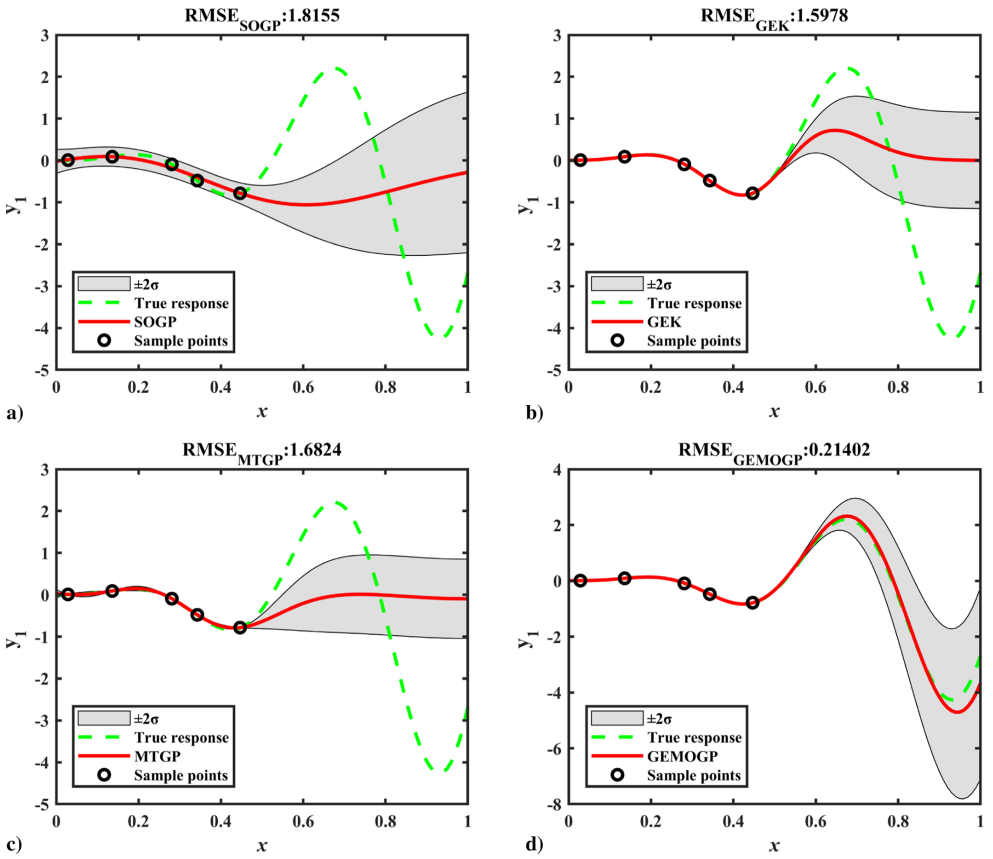


Fig. 7 Predictions of y_1 on the 1-D example for the four different methods: a) SOGP, b) GEK, c) MTGP, and d) GEMOGP.

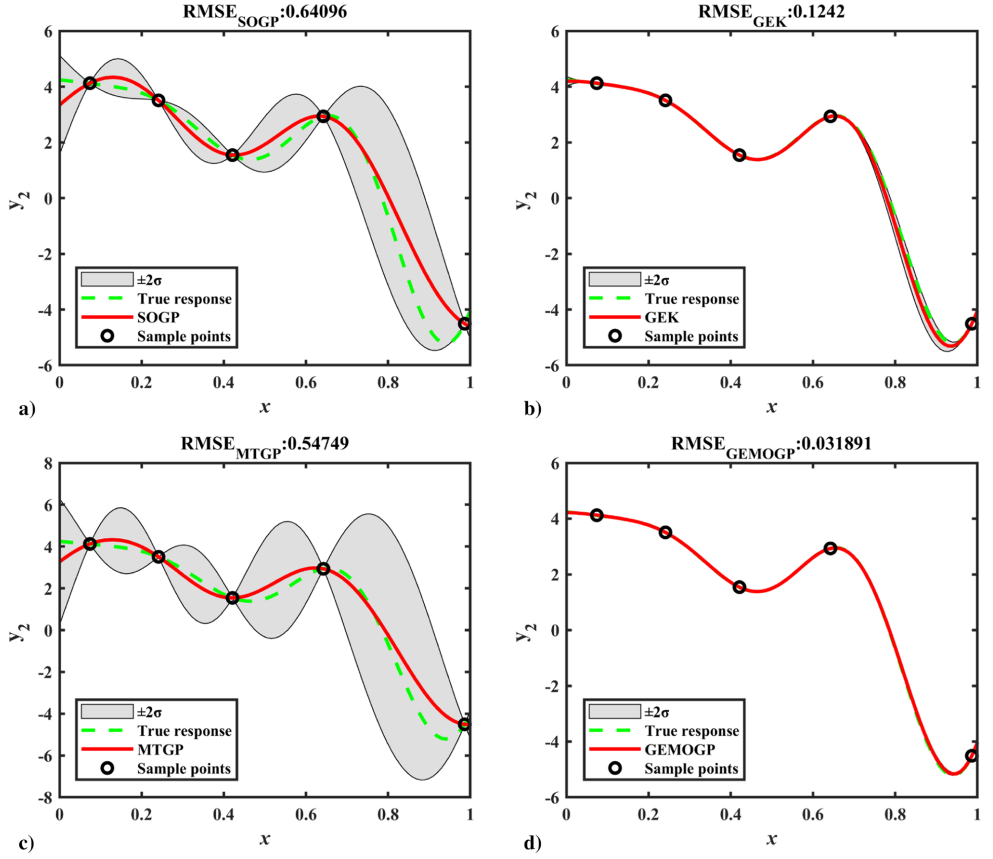


Fig. 8 Predictions of y_2 on the 1-D example for the four different methods: a) SOGP, b) GEK, c) MTGP, and d) GEMOGP.

Table 3 Comparison results for the one-dimensional pedagogical example with partial interval data for y_1

	SOGP		GEK		MTGP		GEMOGP	
Metrics	y_1	y_2	y_1	y_2	y_1	y_2	y_1	y_2
R^2	−0.1895	0.9506	0.0786	0.9981	−0.0214	0.9639	**0.9835**	**0.9999**
RMSE	1.8155	0.6410	1.5978	0.1242	1.6824	0.5475	**0.2140**	**0.0319**
MAE	3.8380	1.7430	4.2796	0.3528	4.1819	1.4190	**0.9849**	**0.0985**

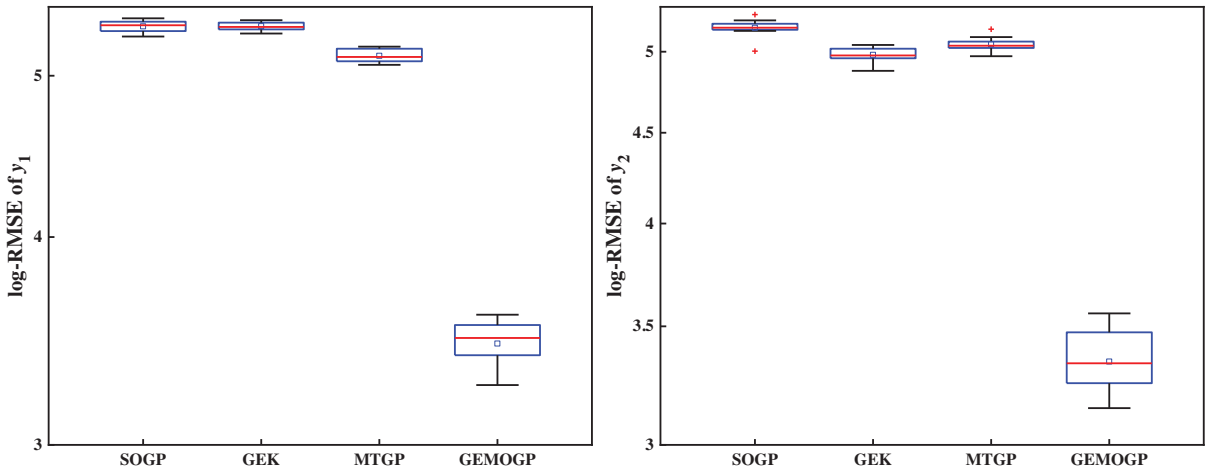


Fig. 9 The log-RMSE values for the four models on the 10-D example with $n_1 = 60$ and $n_2 = 100$.

The "+" symbols represent the outliers. The mean values of the results from the three error metrics are chosen as the representative values, which are summarized in Table 4.

From the results shown in Fig. 9, an intuitive conclusion can be drawn that the proposed GEMOGP model performs best in predicting both outputs with the lowest RMSE values. Because of the limitation of the sample points for this 10-D example, the other three models provide poor predictions, especially for $y_1(x)$. Although the GEK model can obtain more information than the SOGP model, they have similar RMSE values when $n_1 = 60$. Despite this, the proposed model still provides good predictions for $y_1(x)$. This is because of the additional gradient information in all dimensions and information diversity in the 10-D problem, and the GEMOGP model shares the information across outputs. Because the information diversity augments as the dimension increases, it can be inferred that the multi-output modeling approaches become more prominent than the single-output modeling approaches compared with the low-dimensional cases. The GEK model performs better for the predictions of $y_2(x)$ in terms of the global accuracy as the sample size increases to 100. As shown in Table 4, the GEMOGP model has the lowest values for both global and local errors among the four models, indicating that the GEMOGP model fits best for both outputs. It should also be pointed out that, the RMSE and MAE values are still huge, even though the GEMOGP model performs the best with R^2 almost close to 1. This is because the true function values have an order of magnitude of 10^5, whereas the orders of magnitude for the RMSE and MAE values of the GEMOGP model are 10^3 and 10^4, which are much less than the true function values and show accurate predictions for these two functions. For better comparison, some relative error metrics such as normalized root-mean-square error (NRMSE) and normalized maximum absolute error (NMAE) can be adopted for model validation, which are shown in Table C1 the Supplemental Materials.

2. *Influence of Sample Size*

In this part, the influence of different sample sizes on the performance of the proposed model is discussed. For comparison, the heterotopic data are used and the sample sizes of the two outputs are the same. Four groups of sample sizes are tested, namely, 80, 100, 120, and 140.

The three error metrics, RMSE, MAE, and R^2 values, under different sample sizes are shown in Figs. 10–12, respectively. It can be observed that the GEMOGP model performs the best for the two outputs under all tested sample sizes. The other three models provide poor predictions especially when $n = 80$ as they cannot capture the main features of the outputs. As a result, the GEK and MTGP models still provide close predictions compared with the SOGP model, even if they can get more information about the outputs. On the other hand, it can be seen from the figures that the GEMOGP and GEK models yield better predictions with the increase of sample size. This is expected because more information about the outputs can be obtained and the hyperparameters of the model can be inferred more accurately as the number of sampling points increases [4]. Because of the limitation of the sample points, the prediction accuracies of the SOGP and MTGP models change very little regardless of the increase of

Table 4 Comparison results for the four models on the 10-D example

	SOGP		GEK		MTGP		GEMOGP	
Metrics	y_1	y_2	y_1	y_2	y_1	y_2	y_1	y_2
R^2	−0.1993	0.3892	−0.1961	0.7343	0.5496	0.6332	0.9998	0.9998
RMSE	2.27×10^5	1.45×10^5	2.27×10^5	9.59×10^5	1.39×10^5	1.13×10^5	$\mathbf{2.91 \times 10^3}$	$\mathbf{2.30 \times 10^3}$
MAE	1.12×10^6	8.18×10^5	1.11×10^6	6.33×10^5	7.43×10^5	6.31×10^5	$\mathbf{2.17 \times 10^4}$	$\mathbf{1.80 \times 10^4}$

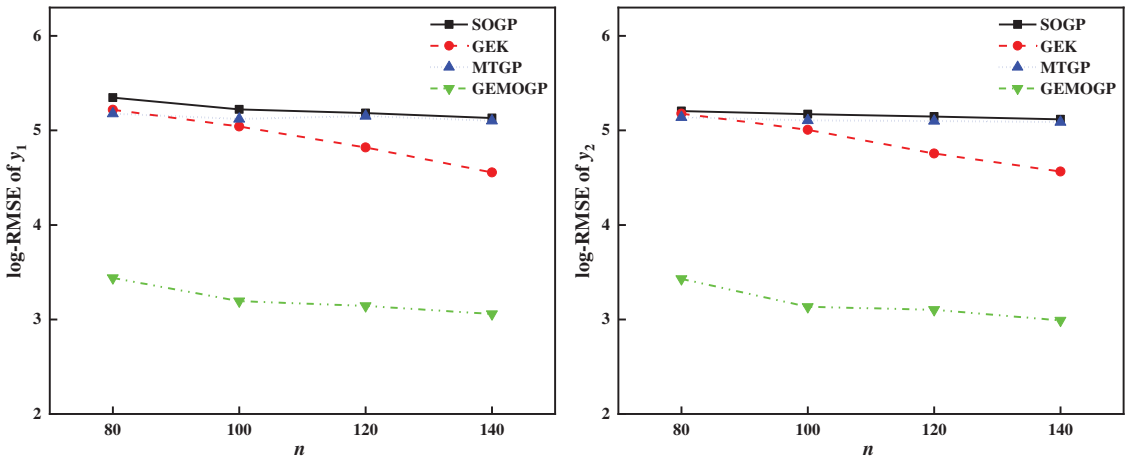


Fig. 10 The log-RMSE values for the four models on the 10-D example under different sample sizes.

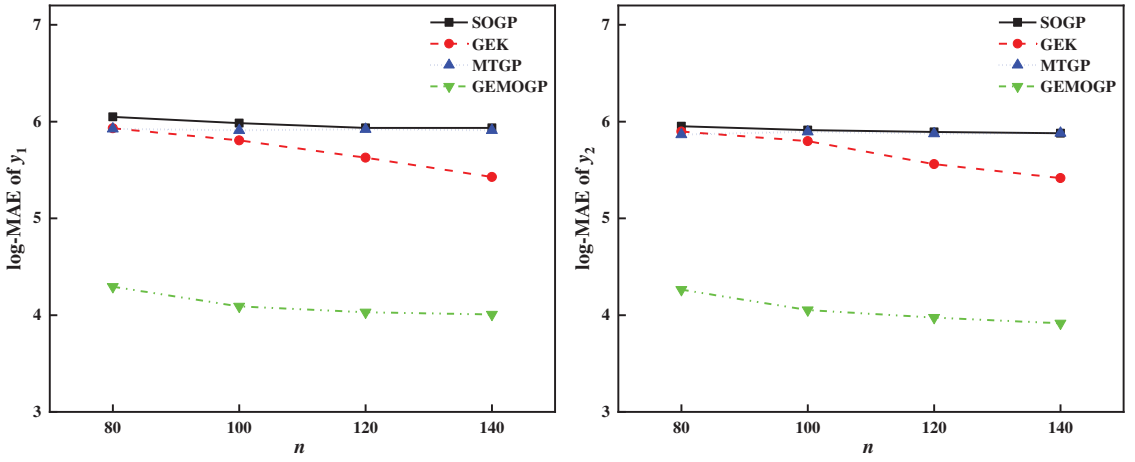


Fig. 11 The log-MAE values for the four models on the 10-D example under different sample sizes.

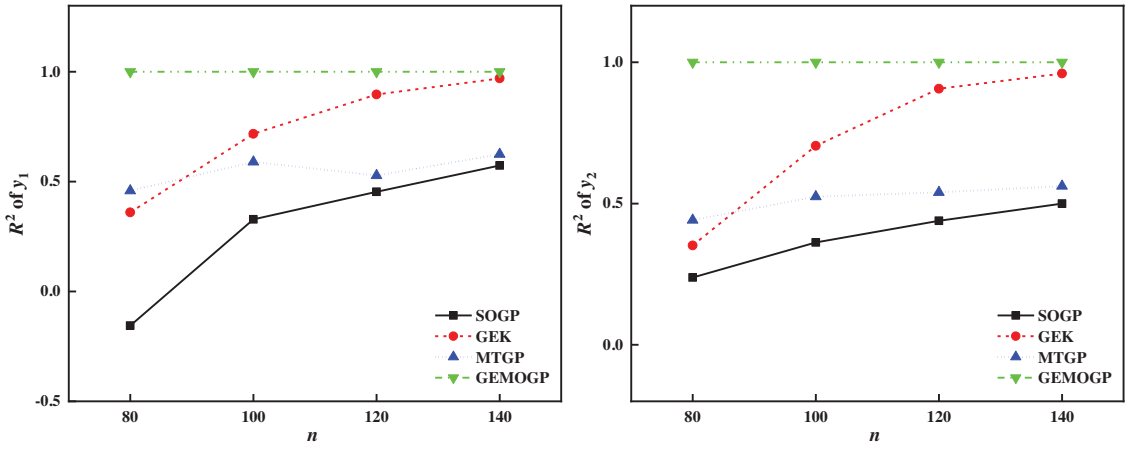
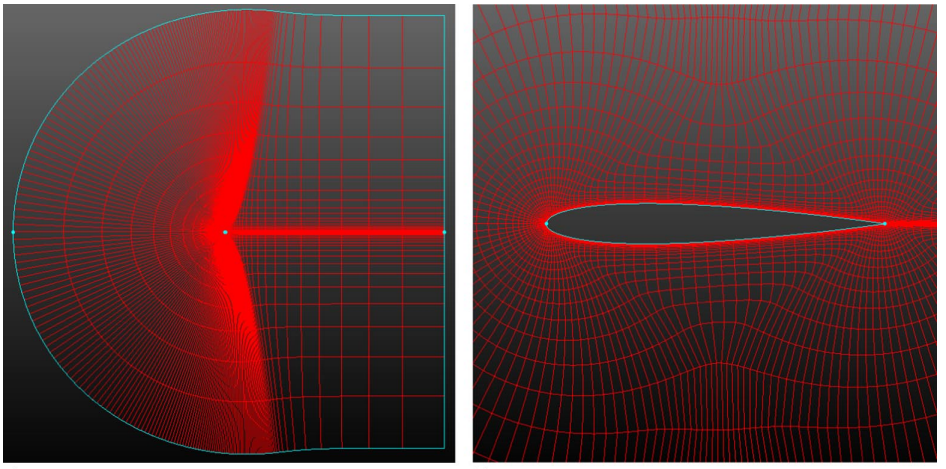


Fig. 12 The R^2 values for the four models on the 10-D example under different sample sizes.

sampling points. It is hard for them to provide good predictions under the currently tested sample sizes. It can be also found that compared with the 1-D case, the proposed model performs more prominent than the other three approaches on the 10-D example. This can be because, as the dimension augments, more information can be obtained and shared across the outputs in each dimension. It may be also interesting to investigate how many samples need to be invested in a GEK model to reach the same accuracy compared with that of the GEMOGP model. For illustration, the Dixon–Price example is used to demonstrate with the dimensionality decreasing to five, which is shown in the Supplemental Materials.

C. Prediction of Aerodynamic Coefficients of a NACA 0012 Airfoil

In this section, the proposed model is applied to predict the aerodynamic coefficients of a NACA 0012 airfoil in subsonic flow. Lift and moment coefficients (C_l and C_m) are considered as two outputs to be approximated. The freestream Mach number (Ma), angle of attack (AoA), and Reynolds number (Re) are fixed to $Ma = 0.3$, $AoA = 10.0254$ deg, and $Re = 6.0 \times 10^6$, respectively. The Hicks–Henne bump functions with a total of 18 variables are used for the geometrical parameterization of the airfoil. In this case, a new airfoil shape is yielded by adding disturbations Δ_u and Δ_l on the upper and lower surfaces of the baseline shape, where



a) b)

Fig. 13 Far-field (a) and zoom view (b) of the computational mesh of the NACA 0012 airfoil.

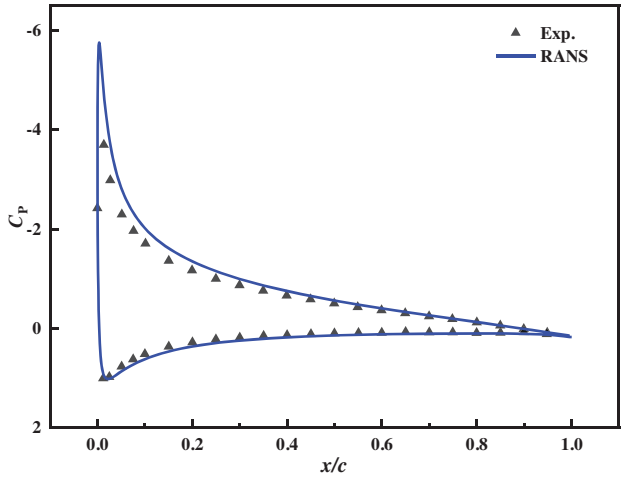


Fig. 14 Validation of the flow solver by comparison of the calculated pressure distribution and experimental data of the NACA 0012 airfoil ($Ma = 0.3$, $AoA = 10.0254$ deg, $Re = 6.0 \times 10^6$).

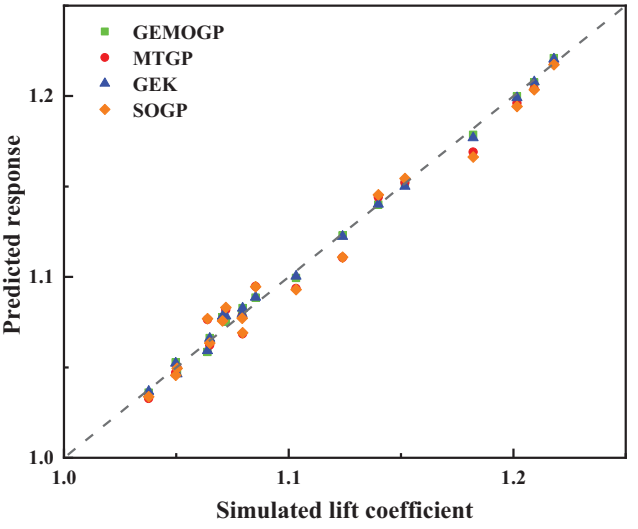


Fig. 16 Comparison of simulated lift coefficient and predicted values at test points (some data are overlapped).

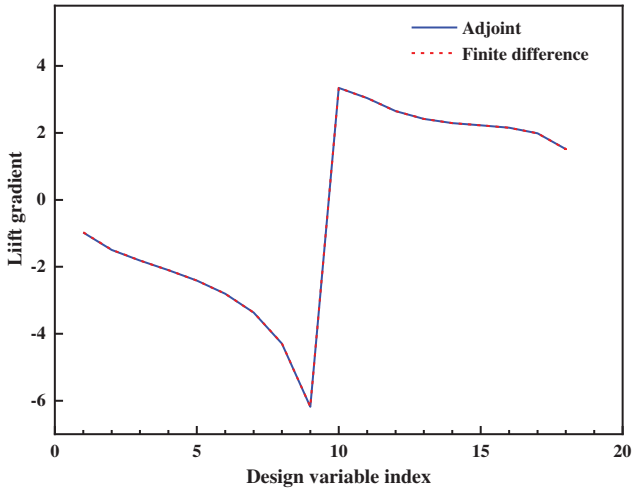


Fig. 15 Validation of the adjoint solver by comparing computed lift gradients with those by the finite difference method.

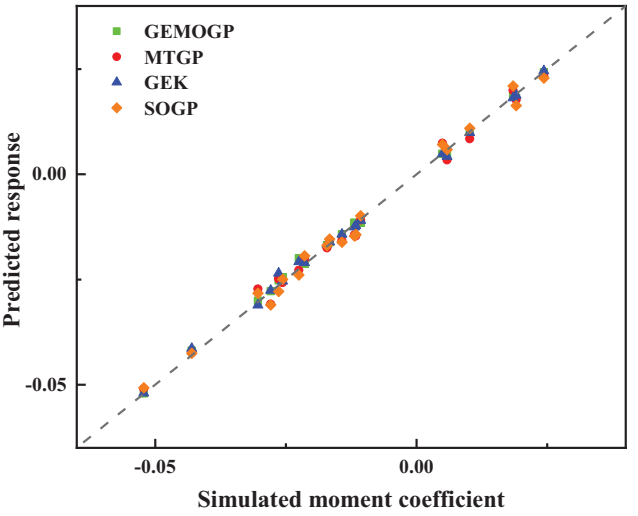


Fig. 17 Comparison of simulated moment coefficient and predicted values at test points (some data are overlapped).

Table 5 Comparison results for the predictions of aerodynamic coefficients for a NACA 0012 airfoil

	SOGP		GEK		MTGP		GEMOGP	
Metrics	C_l	C_m	C_l	C_m	C_l	C_m	C_l	C_m
R^2	0.9873	0.9918	0.9974	0.9976	0.9887	0.9922	0.9978	0.9983
RMSE	7.81×10^{-3}	1.83×10^{-3}	3.53×10^{-3}	9.82×10^{-4}	7.37×10^{-3}	1.78×10^{-3}	3.25×10^{-3}	8.23×10^{-4}
MAE	1.58×10^{-2}	3.12×10^{-3}	7.67×10^{-3}	2.90×10^{-3}	1.33×10^{-2}	3.09×10^{-3}	7.10×10^{-3}	2.50×10^{-3}

$$\Delta_u = \sum_{i=1}^{9} \delta_{ui} f_i(x) \qquad \delta_{ui} \in [-0.01, 0.01]$$

$$\Delta_l = -\sum_{i=1}^{9} \delta_{li} f_i(x) \qquad \delta_{li} \in [-0.01, 0.01] \tag{31}$$

are the total deformations on the upper and lower surfaces, with

$$f_i(x) = \sin^3(\pi x^{e(i)}) \tag{32}$$

$$e(i) = \log(0.5)/\log(x_i) \qquad i = 1, \ldots, 9 \tag{33}$$

where $x_i \in [0, 1]$ is the location of the function maximum, which is uniformly distributed at [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] in this problem. The amplitudes of bumps $\delta_{ui}(i = 1, \ldots, 9)$ and $\delta_{li}(i = 1, \ldots, 9)$ are the design variables. The original computational mesh consists of 14,576 points, as shown in Fig. 13. For comparison, the isotopic data are used to construct the four surrogate models. A total of 40 sampling points are generated by *lhsdesign* with *maximin* criterion for each output, and 20 test points are randomly selected to compare the performance of the four surrogate models. The gradients with respect to design variables are calculated by the adjoint method. All the aerodynamic computations (lift and drag coefficients, gradients with respect to design variables) are solved by SU2 [45], a suite of open-source software tools written in C++ for the numerical solution of partial differential equations (PDEs). The flow solver is validated by comparing the computed pressure distributions with the wind tunnel experiment, as shown in Fig. 14. One can observe that the differences between the CFD simulation results and the experimental data are within a reasonable range. Figure 15 shows the comparison results of the computed lift gradients with those by the finite difference method. One can also observe that the gradients obtained by the adjoint method are in good agreement with those by the finite difference.

The comparisons of the simulation validation data and predicted values of C_l and C_m are illustrated in Figs. 16 and 17, respectively. It can be seen that the GEMOGP and GEK models provide comparable predictions for both C_l and C_m, and perform better than the other two models. The SOGP model performs the worst among the four surrogate models. Three error metrics, R^2, RMSE, and MAE, of the four models are listed in Table 5. It can be concluded from Table 5 that the proposed model outperforms the other three models on the predictions for both C_l and C_m with the lowest RMSE and MAE values. The RMSE values for C_l and C_m are reduced by 7.9 and 16.2% using the GEMOGP model compared with those by using the GEK model, whereas compared with those by using the MTGP model, the RMSE values are reduced by 55.9 and 53.8%, respectively. A conclusion can be drawn that both the information transferred across outputs and the information of the gradient are helpful to the model accuracy.

On the other hand, the CPU time for the GEK and GEMOGP modeling, as well as the CFD time, is recorded. Results show that the training time of the GEK and GEMOGP models is about 815 and 1,632 s, respectively, whereas the total CFD time is about 81,640 s, which is much larger than the modeling time. Therefore, it should be pointed out that although the computational cost for training the proposed model is the most expensive one among the four tested models, the cost of training a model is much less than the expensive simulations. Overall, for design problems involving computationally expensive simulations, it is worthwhile applying the proposed model to decrease the simulation costs at the expense of more training costs.

V. Conclusions

In this paper, a GEMOGP modeling approach assisted by gradient information is proposed for multi-output prediction. In the multi-output scenario, the proposed model can be applied if the gradient information at the sample locations can be achieved. The GEMOGP model combines the observed responses with gradient information as additional training data to make the most use of sample information so as to provide more accurate predictions. Besides, the correlation information across multiple outputs can also be captured to enhance the prediction accuracy of the model. The performance of the proposed approach is compared with three other GP-based approaches (SOGP, GEK, and MTGP) using two analytical cases and an engineering example. The observations are summarized as follows:

1) If the outputs have some correlations, for the same sample points, the proposed GEMOGP approach outperforms the other three approaches dramatically in terms of both local and global accuracy.

2) The proposed GEMOGP approach performs more prominent than the other three approaches when using the heterotopic training data.

3) In the NACA 0012 airfoil case, the RMSE value of C_l is reduced by 7.9% using the GEMOGP model compared with that of the GEK, and 55.9% compared with that of the MTGP, whereas for the RMSE value of C_m the obtained value by using the GEMOGP model is reduced by 16.2% compared with that of the GEK, and 53.8% compared with that of the MTGP.

It should be pointed out that incorporating gradient information at the sample locations in the GEMOGP model does require additional computational cost, whereas this cost can be ignored when comparing with the expensive simulations. Besides, the "curse of dimensionality" still exists in the proposed model. As a part of future work, combining the proposed model with dimension reduction techniques, e.g., proper orthogonal decomposition [46,47], will be investigated to alleviate this problem.

Acknowledgments

This research has been supported by the National Natural Science Foundation of China under Grant No. 51805179, No. 51775203, and No. 51721092; the China Postdoctoral Science Foundation under Grant No. 2020M682396; and the Research Funds of the Maritime Defense Technologies Innovation under Grant No. YT19201901.

References

[1] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, 1989, pp. 409–423.

[2] Mullur, A. A., and Messac, A., "Extended Radial Basis Functions: More Flexible and Effective Metamodeling," *AIAA Journal*, Vol. 43, No. 6, 2005, pp. 1306–1315. https://doi.org/10.2514/1.11292

[3] Smola, A. J., and Schölkopf, B., "A Tutorial on Support Vector Regression," *Statistics and Computing*, Vol. 14, No. 3, 2004, pp. 199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88

[4] Williams, C. K., and Rasmussen, C. E., *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, MA, 2006, Chap. 2.

[5] Keshavarzzadeh, V., Kirby, R. M., and Narayan, A., "Stress-Based Topology Optimization Under Uncertainty via Simulation-Based Gaussian Process," *Computer Methods in Applied Mechanics and Engineering*, Vol. 365, June 2020, Paper 112992. https://doi.org/10.1016/j.cma.2020.112992

[6] Hu, J., Zhou, Q., McKeand, A., Xie, T., and Choi, S.-K., "A Model Validation Framework Based on Parameter Calibration Under Aleatory and Epistemic Uncertainty," *Structural and Multidisciplinary Optimi-*

zation, Vol. 63, No. 2, 2021, pp. 645–660.
https://doi.org/10.1007/s00158-020-02715-z

[7] Durichen, R., Pimentel, M. A., Clifton, L., Schweikard, A., and Clifton, D. A., "Multitask Gaussian Processes for Multivariate Physiological Time-Series Analysis," *IEEE Transactions on Biomedical Engineering*, Vol. 62, No. 1, 2015, pp. 314–322.
https://doi.org/10.1109/TBME.2014.2351376

[8] Lu, J., Zhan, Z., Apley, D. W., and Chen, W., "Uncertainty Propagation of Frequency Response Functions Using a Multi-Output Gaussian Process Model," *Computers & Structures*, Vol. 217, June 2019, pp. 1–17.
https://doi.org/10.1016/j.compstruc.2019.03.009Get

[9] Liu, X., Zhu, Q., and Lu, H., "Modeling Multiresponse Surfaces for Airfoil Design with Multiple-Output-Gaussian-Process Regression," *Journal of Aircraft*, Vol. 51, No. 3, 2014, pp. 740–747.
https://doi.org/10.2514/1.C032465

[10] Journel, A. G., and Huijbregts, C. J., *Mining Geostatistics*, Academic Press, London, 1978.

[11] Goovaerts, P., *Geostatistics for Natural Resources Evaluation*, Oxford Univ. Press, Orford, NY, 1997, Chap. 4.

[12] Seeger, M., Teh, Y.-W., and Jordan, M., "Semiparametric Latent Factor Models," 2005.

[13] Bonilla, E. V., Chai, K. M. A., and Williams, C. K. I., "Multi-Task Gaussian Process Prediction," *Proceedings of the 20th International Conference on Neural Information Processing Systems*, Curran Associates Inc., Vancouver, British Columbia, Canada, 2007, pp. 153–160.

[14] Boyle, P., and Frean, M., "Dependent Gaussian Processes," *Proceedings of the 17th International Conference on Neural Information Processing Systems*, MIT Press, Vancouver, British Columbia, Canada, 2004, pp. 217–224.

[15] Alvarez, M., and Lawrence, N. D., "Sparse Convolved Gaussian Processes for Multi-Output Regression," *Proceedings of the 21st International Conference on Neural Information Processing Systems*, Curran Associates Inc., Vancouver, British Columbia, Canada, 2008, pp. 57–64.

[16] Lawrence, N. D., Rosasco, L., and Álvarez, M. A., "Kernels for Vector-Valued Functions: A Review," *Foundations and Trends® in Machine Learning*, Vol. 4, No. 3, 2012, pp. 195–266.
https://doi.org/10.1561/2200000036

[17] Liu, H., Cai, J., and Ong, Y.-S., "Remarks on Multi-Output Gaussian Process Regression," *Knowledge-Based Systems*, Vol. 144, March 2018, pp. 102–121.
https://doi.org/10.1016/j.knosys.2017.12.034

[18] Giles, M. B., and Pierce, N. A., "An Introduction to the Adjoint Approach to Design," *Flow, Turbulence and Combustion*, Vol. 65, Nos. 3–4, 2000, pp. 393–415.
https://doi.org/10.1023/A:1011430410075

[19] Neidinger, R. D., "Introduction to Automatic Differentiation and MATLAB Object-Oriented Programming," *SIAM Review*, Vol. 52, No. 3, 2010, pp. 545–563.
https://doi.org/10.1137/080743627

[20] Martins, J. R., Alonso, J. J., and Reuther, J. J., "A Coupled-Adjoint Sensitivity Analysis Method for High-Fidelity Aero-Structural Design," *Optimization and Engineering*, Vol. 6, No. 1, 2005, pp. 33–62.
https://doi.org/10.1023/B:OPTE.0000048536.47956.62

[21] Mavriplis, D. J., "Discrete Adjoint-Based Approach for Optimization Problems on Three-Dimensional Unstructured Meshes," *AIAA Journal*, Vol. 45, No. 4, 2007, pp. 741–750.
https://doi.org/10.2514/1.22743

[22] Cheylan, I., Fritz, G., Ricot, D., and Sagaut, P., "Shape Optimization Using the Adjoint Lattice Boltzmann Method for Aerodynamic Applications," *AIAA Journal*, Vol. 57, No. 7, 2019, pp. 2758–2773.
https://doi.org/10.2514/1.J057955

[23] Morris, M. D., Mitchell, T. J., and Ylvisaker, D., "Bayesian Design and Analysis of Computer Experiments: Use of Derivatives in Surface Prediction," *Technometrics*, Vol. 35, No. 3, 1993, pp. 243–255.
https://doi.org/10.1080/00401706.1993.10485320

[24] Chen, L., Qiu, H., Gao, L., Jiang, C., and Yang, Z., "Optimization of Expensive Black-Box Problems via Gradient-Enhanced Kriging," *Computer Methods in Applied Mechanics and Engineering*, Vol. 362, April 2020, Paper 112861.
https://doi.org/10.1016/j.cma.2020.112861

[25] Ulaganathan, S., Couckuyt, I., Dhaene, T., Degroote, J., and Laermans, E., "Performance Study of Gradient-Enhanced Kriging," *Engineering with Computers*, Vol. 32, No. 1, 2016, pp. 15–34.
https://doi.org/10.1007/s00366-015-0397-y

[26] de Baar, J. H., Dwight, R. P., and Bijl, H., "Improvements to Gradient-Enhanced Kriging Using a Bayesian Interpretation," *International Journal for Uncertainty Quantification*, Vol. 4, No. 3, 2014, pp. 205–223.
https://doi.org/10.1615/Int.J.UncertaintyQuantification.2013006809

[27] Han, Z.-H., Zhang, Y., Song, C.-X., and Zhang, K.-S., "Weighted Gradient-Enhanced Kriging for High-Dimensional Surrogate Modeling and Design Optimization," *AIAA Journal*, Vol. 55, No. 12, 2017, pp. 4330–4346.
https://doi.org/10.2514/1.J055842

[28] Chung, H.-S., and Alonso, J., "Using Gradients to Construct Cokriging Approximation Models for High-Dimensional Design Optimization Problems," *40th AIAA Aerospace Sciences Meeting & Exhibit*, AIAA Paper 2002-0317, 2002.

[29] Chung, H. S., and Alonso, J., "Design of a Low-Boom Supersonic Business Jet Using Cokriging Approximation Models," *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA Paper 2002-5598, 2002.

[30] Zimmermann, R., "On the Maximum Likelihood Training of Gradient-Enhanced Spatial Gaussian Processes," *SIAM Journal on Scientific Computing*, Vol. 35, No. 6, 2013, pp. A2554–A2574.
https://doi.org/10.1137/13092229X

[31] Liu, W., and Batill, S., "Gradient-Enhanced Response Surface Approximations Using Kriging Models," *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA Paper 2002-5456, 2002.

[32] Laurent, L., Le Riche, R., Soulier, B., and Boucard, P.-A., "An Overview of Gradient-Enhanced Metamodels with Applications," *Archives of Computational Methods in Engineering*, Vol. 26, No. 1, 2019, pp. 61–106.
https://doi.org/10.1007/s11831-017-9226-3

[33] Wackernagel, H., *Multivariate Geostatistics: An Introduction with Applications*, Springer Science & Business Media, Springer-Verlag, Berlin, 2013, p. 158.

[34] Rakitsch, B., Lippert, C., Borgwardt, K. M., and Stegle, O., "It Is All in the Noise: Efficient Multi-Task Gaussian Process Inference with Structured Residuals," *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, Curran Associates, Inc., Lake Tahoe, Nevada, 2013, pp. 1466–1474.

[35] Cohn, T., and Specia, L., "Modelling Annotator Bias with Multi-Task Gaussian Processes: An Application to Machine Translation Quality Estimation," *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1, Assoc. for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 32–42.

[36] Osborne, M. A., Roberts, S. J., Rogers, A., and Jennings, N. R., "Real-Time Information Processing of Environmental Sensor Network Data Using Bayesian Gaussian Processes," *ACM Transactions on Sensor Networks*, Vol. 9, No. 1, 2012, pp. 1–32.
https://doi.org/10.1145/2379799.2379800

[37] Han, Z., "Improving Adjoint-Based Aerodynamic Optimization via Gradient-Enhanced Kriging," *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, AIAA Paper 2012-0670, 2012.

[38] Wang, B., and Chen, T., "Gaussian Process Regression with Multiple Response Variables," *Chemometrics and Intelligent Laboratory Systems*, Vol. 142, March 2015, pp. 159–165.
https://doi.org/10.1016/j.chemolab.2015.01.016

[39] Martin, J. D., and Simpson, T. W., "Use of Kriging Models to Approximate Deterministic Computer Models," *AIAA Journal*, Vol. 43, No. 4, 2005, pp. 853–863.
https://doi.org/10.2514/1.8650

[40] Zhang, J., and Sanderson, A. C., "JADE: Adaptive Differential Evolution with Optional External Archive," *IEEE Transactions on Evolutionary Computation*, Vol. 13, No. 5, 2009, pp. 945–958.
https://doi.org/10.1109/TEVC.2009.2014613

[41] Martins, J. R., Kroo, I., and Alonso, J., "An Automated Method for Sensitivity Analysis Using Complex Variables," *38th Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2000-0689, 2000.

[42] Wang, G. G., and Shan, S., "Review of Metamodeling Techniques in Support of Engineering Design Optimization," *Journal of Mechanical Design*, Vol. 129, No. 4, 2007, pp. 415–426.
https://doi.org/10.1115/1.2429697

[43] Liu, H., Ong, Y.-S., Cai, J., and Wang, Y., "Cope with Diverse Data Structures in Multi-Fidelity Modeling: A Gaussian Process Method," *Engineering Applications of Artificial Intelligence*, Vol. 67, Jan. 2018, pp. 211–225.
https://doi.org/10.1016/j.engappai.2017.10.008

[44] Cai, X., Qiu, H., Gao, L., and Shao, X., "Metamodeling for High Dimensional Design Problems by Multi-Fidelity Simulations," *Structural and Multidisciplinary Optimization*, Vol. 56, No. 1, 2017, pp. 151–166.

[45] Economon, T. D., Palacios, F., Copeland, S. R., Lukaczyk, T. W., and Alonso, J. J., "SU2: An Open-Source Suite for Multiphysics

Simulation and Design," *AIAA Journal*, Vol. 54, No. 3, 2016, pp. 828–846.
https://doi.org/10.2514/6.2013-287

[46] Rokita, T., and Friedmann, P. P., "Multifidelity Cokriging for High-Dimensional Output Functions with Application to Hypersonic Airloads Computation," *AIAA Journal*, Vol. 56, No. 8, 2018, pp. 3060–3070.
https://doi.org/10.2514/1.J056620

[47] Swischuk, R., Kramer, B., Huang, C., and Willcox, K., "Learning Physics-Based Reduced-Order Models for a Single-Injector Combustion Process," *AIAA Journal*, Vol. 58, No. 6, 2020, pp. 2658–2672.
https://doi.org/10.2514/1.J058943

K. E. Willcox
Associate Editor