Remarks on Multi-Output Gaussian Process Regression

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

[a] *Rolls-Royce@NTU Corporate Lab, Nanyang Technological University, Singapore 637460*
[b] *School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798*
[c] *Data Science and Artificial Intelligence Research Center, Nanyang Technological University, Singapore 639798*

Abstract

Multi-output regression problems have extensively arisen in modern engineering community. This article investigates the state-of-the-art multi-output Gaussian processes (MOGPs) that can transfer the knowledge across related outputs in order to improve prediction quality. We classify existing MOGPs into two main categories as (1) symmetric MOGPs that improve the predictions for all the outputs, and (2) asymmetric MOGPs, particularly the multi-fidelity MOGPs, that focus on the improvement of high fidelity output via the useful information transferred from related low fidelity outputs. We review existing symmetric/asymmetric MOGPs and analyze their characteristics, e.g., the covariance functions (separable or non-separable), the modeling process (integrated or decomposed), the information transfer (bidirectional or unidirectional), and the hyperparameter inference (joint or separate). Besides, we assess the performance of ten representative MOGPs thoroughly on eight examples in symmetric/asymmetric scenarios by considering, e.g., different training data (heterotopic or isotopic), different training sizes (small, moderate and large), different output correlations (low or high), and different output sizes (up to four outputs). Based on the qualitative and quantitative analysis, we give some recommendations regarding the usage of MOGPs and highlight potential research directions.

Keywords: multi-output Gaussian process, symmetric/asymmetric MOGP, multi-fidelity, output correlation, knowledge transfer

1. Introduction

Computer simulators, e.g., computational fluid dynamics (CFD) and finite element analysis (FEA), have gained popularity in many scientific fields to simulate various physical problems. For computationally expensive simulators, we

*Corresponding author
Email addresses: htliu@ntu.edu.sg (Haitao Liu), ASJFCai@ntu.edu.sg (Jianfei Cai), ASYSOng@ntu.edu.sg (Yew-Soon Ong)

usually employ surrogates to approximate the input-output relationship in order to relieve computational budget [1, 2, 3]. As a statistical surrogate model that provides not only the predictions but also the relevant uncertainty, Gaussian process (GP) has been gaining widespread applications, e.g., small- or large-scale regression [4, 5, 6], dimensionality reduction [7], Bayesian optimization [8], uncertainty quantification [9] and time-series analysis [10].

Typical GPs are usually designed for single-output scenarios wherein the output is a scalar. However, the multi-output problems have arisen in various fields, e.g., environmental sensor networks [11], robot inverse dynamics [12], multivariate physiological time-series analysis [13], structural design [14], and aircraft design [15]. Suppose that we attempt to approximate T outputs $\{f_t\}_{1\leq t\leq T}$, one intuitive idea is to use the single-output GP (SOGP) to approximate them individually using the associated training data $\mathcal{D}_t = \{X_t, \mathbf{y}_t\}$, see Fig. 1(a). Considering that the outputs are correlated in some way, modeling them individually may result in the loss of valuable information. Hence, an increasing diversity of engineering applications are embarking on the use of multi-output GP (MOGP), which is conceptually depicted in Fig. 1(b), for surrogate modeling.

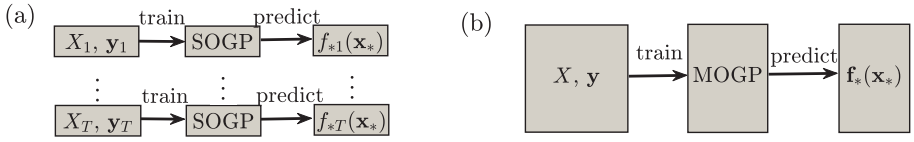

Figure 1: Illustration of (a) the SOGP and (b) the MOGP.

The study of MOGP has a long history and is known as multivariate Kriging or Co-Kriging [16, 17, 18, 19] in the geostatistic community; it also overlaps with the broad field of multi-task learning [20, 21] and transfer learning [22, 23] of the machine learning community. The MOGP handles problems with the basic assumption that the outputs are correlated in some way. Hence, a key issue in MOGP is to *exploit the output correlations such that the outputs can leverage information from one another* in order to provide more accurate predictions in comparison to modeling them individually.

Existing MOGPs can in general be classified into two categories: (1) *symmetric* MOGPs and (2) *asymmetric* MOGPs. Symmetric MOGPs use a *symmetric dependency structure* to capture the output correlations and approximate the T outputs simultaneously. Therefore, these MOGPs usually have an *integrated* modeling process, i.e., fusing all the information in an entire covariance matrix, which leads to *bidirectional* information transfer between the outputs. Typically, the symmetric MOGPs attempt to improve the predictions of all the outputs in symmetric scenarios, where the outputs are of equal importance and have roughly equivalent training information.

On the contrary, asymmetric MOGPs, which have an *asymmetric dependency structure* specifically designed for asymmetric scenarios, target to enhance the *primary* output predictions by transferring useful knowledge from other re-

lated *secondary* outputs [1]. The basic assumption is that the primary output has a few training points, but the secondary outputs, also denoted as source domains in transfer learning [24, 25, 26], usually have sufficient training points. Here, we particularly restrict ourselves to a *hierarchical* asymmetric scenario where the simulator for the physics-based problem of interests has multiple levels of fidelity. Regarding them as different outputs, the version with the highest fidelity is the primary output, which has been deem to give the most accurate predictions but is most time-consuming; whereas the simple and fast versions with declining fidelities provide coarse predictions, which however include the main features of the engineering problem and thus are useful for preliminary exploration. This kind of asymmetric multi-output modeling is often referred to as *multi-fidelity modeling* or *variable fidelity modeling* [27, 28, 29].

This article intends to (1) review and analyze the characteristics and differences of the state-of-the-art symmetric/asymmetric MOGPs, (2) investigate the potential of MOGPs over the typical SOGP on symmetric/asymmetric examples, and (3) give some recommendations regarding the usage of MOGPs.

The remainder of this article is organized as follows. Section 2 introduces the general single-/multi-output GP modeling framework. Thereafter, the existing symmetric and asymmetric MOGPs are reviewed and analyzed in Section 3 and Section 4, respectively. Section 5 further discusses the inference methods as well as the computational considerations to implement these MOGPs in practice. Subsequently, Section 6 investigates the performance and characteristics of symmetric MOGPs on four symmetric examples. Moreover, Section 7 studies the asymmetric/symmetric MOGPs on four asymmetric examples. Last, some concluding remarks are provided in Section 8.

2. Single-/Multi-output Gaussian process modeling framework

In the multi-output scenario, assume that $X = \{\mathbf{x}_{t,i}|t = 1, \cdots, T; i = 1, \cdots, n_t\}$ and $\mathbf{y} = \{y_{t,i} = y_t(\mathbf{x}_{t,i})|t = 1, \cdots, T; i = 1, \cdots, n_t\}$ are the collection of training points and associated observations for T outputs $\{f_t\}_{1 \le t \le T}$. Suppose that $N = \sum_{t=1}^{T} n_t$, the matrix $X \in R^{N \times d}$ has T blocks with the t-th block $X_t = \{\mathbf{x}_{t,1}, \cdots, \mathbf{x}_{t,n_t}\}^{\mathsf{T}}$ corresponding to the training set for output f_t; the vector $\mathbf{y} \in R^{N \times 1}$ also has T components with $\mathbf{y}_t = \{y_{t,1}, \cdots, y_{t,n_t}\}^{\mathsf{T}}$ corresponding to the observations of f_t at X_t.

Given the training data $\mathcal{D} = \{X, \mathbf{y}\}$ for T outputs, the task is to learn a MOGP model as

$$\text{MOGP} : \Omega_d \to \Omega_{f_1} \times \cdots \times \Omega_{f_T}$$

where Ω_d represents the d-dimensional input space, and Ω_{f_t} represents the output space for $f_t(\mathbf{x})$. In this article, we assume that all the T outputs share the

[1] Though symmetric MOGPs are available in asymmetric scenarios (see the illustration examples in [13]), they may be not so effective as the particularly designed asymmetric MOGPs.

same input space [2]. Besides, for the training sets, we consider two configurations below:

- *Heterotopic data*. It means the T outputs have different training sets, i.e., $X_1 \neq \cdots \neq X_T$. The heterotopic data often occurs in the scenario where the T output responses at a point $\mathbf{x}$ can be obtained by separate simulations.

- *Isotopic data*. It indicates the T outputs have the same training set, i.e., $X_1 = \cdots = X_T = \overline{X}$. The isotopic data often occurs in the scenario where the T output responses at a point $\mathbf{x}$ can be obtained simultaneously through a single simulation.

For the modeling of the outputs, this article considers two scenarios below:

- *Symmetric scenario*. In this scenario, the T outputs are of equal importance and have the same number of training points, i.e., $n_1 = \cdots = n_T = n$. This scenario attempts to improve the predictions of all the outputs and has been popularly studied in multi-output regression [30]. Both the heterotopic data and the isotopic data are available in this scenario, and the symmetric MOGPs can be used here.

- *Asymmetric scenario*. In this article, it particularly refers to the hierarchical multi-fidelity scenario where $n_1 > \cdots > n_T$. This scenario attempts to improve the predictions of the expensive high fidelity (HF) output f_T by transferring information from the inexpensive low fidelity (LF) outputs $\{f_t\}_{1 \leq t \leq T-1}$. Note that only the heterotopic training data occurs in this scenario, and both the symmetric and asymmetric MOGPs can be used here.

Throughout the section, we first introduce the typical single-output GP modeling framework, followed by the general multi-output GP modeling framework.

2.1. Single-output Gaussian process

Typical single-output GP (SOGP) attempts to approximate the target output $f(\mathbf{x})$ where $\mathbf{x} \in R^d$ by interpreting it 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 $m(\mathbf{x})$, which is usually taken as zero without loss of generality, and the covariance function $k(\mathbf{x}, \mathbf{x}')$. The well-known squared exponential (SE) covariance function [4], which is infinitely differentiable and smooth, is expressed as

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

[2] Some works have considered the scenario where the outputs have different input spaces, see the review paper [22].

where the signal variance σ_f^2 represents an output scale amplitude; the diagonal matrix $P \in R^{d\times d}$ contains the characteristic length scales $\{l_i^2\}_{1\le i\le d}$ that represent the oscillation frequencies along different directions.

In many realistic scenarios, we only have the observation of the exact function value as

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

where the independent and identically distributed (iid) noise $\epsilon \sim \mathcal{N}(0, \sigma_s^2)$ accounts for the measurement errors, the modeling errors, or the manufacturing tolerances. The consideration of noise in GP is beneficial for numerical stability [31, 32] and better statistical properties [33] [3].

Suppose that we have a set of training points $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}^\mathsf{T}$ in the domain Ω_d and the associated output observations $\mathbf{y} = \{y(\mathbf{x}_1), \cdots, y(\mathbf{x}_n)\}^\mathsf{T}$. Since GP is a stochastic process wherein any finite subset of random variables follows a joint Gaussian distribution, the joint prior distribution of the observations $\mathbf{y}$ together with $f(\mathbf{x}_*)$ at a test point $\mathbf{x}_*$ is

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

where $K(X,X) \in R^{n\times n}$ is the symmetric and positive semi-definite (PSD) covariance matrix with the element $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. By conditioning the joint Gaussian prior distribution on $\mathbf{y}$, the posterior distribution of $f(\mathbf{x}_*)$ is analytically derived as

$$f(\mathbf{x}_*)|X, \mathbf{y}, \mathbf{x}_* \sim \mathcal{N}(\hat{f}(\mathbf{x}_*), \sigma^2(\mathbf{x}_*)). \tag{5}$$

The prediction mean $\hat{f}(\mathbf{x}_*)$ and prediction variance $\sigma^2(\mathbf{x}_*)$ are respectively given as

$$\hat{f}(\mathbf{x}_*) = \mathbf{k}_*^\mathsf{T}[K(X,X) + \sigma_s^2 I]^{-1}\mathbf{y}, \tag{6a}$$

$$\sigma^2(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\mathsf{T}[K(X,X) + \sigma_s^2 I]^{-1}\mathbf{k}_*, \tag{6b}$$

where $\mathbf{k}_* = K(X, \mathbf{x}_*) \in R^{n\times 1}$ denotes the covariance between the n training points and the test point $\mathbf{x}_*$. Note that the prediction variance of $y(\mathbf{x}_*)$ is $\sigma^2(\mathbf{x}_*) + \sigma_s^2$.

To use (6a) and (6b) for prediction, we need to infer the hyperparametes $\boldsymbol{\theta}$ in the covariance function k and the noise process ϵ by minimizing the negative log marginal likelihood (NLML) as

$$\boldsymbol{\theta}_{\text{opt}} = \arg\min_{\boldsymbol{\theta}} \text{NLML}, \tag{7}$$

where

$$\begin{aligned} \text{NLML} &= -\log p(\mathbf{y}|X, \boldsymbol{\theta}) \\ &= \frac{1}{2}\mathbf{y}^\mathrm{T}[K(X,X) + \sigma_s^2 I]^{-1}\mathbf{y} + \frac{1}{2}\log|K(X,X) + \sigma_s^2 I| + \frac{n}{2}\log 2\pi. \end{aligned} \tag{8}$$

[3] In the so-called "deterministic" scenario, people often deliberately omit the noise ϵ to confine the GP model to an interpolator. However, Gramacy et al. [33] argued that this is a too narrow and statistically inefficient way to approximate the underlying function.

Alternatively, problem (7) can be solved by the efficient gradient descent algorithm via the partial derivatives of the marginal likelihood w.r.t. the hyperparameters $\boldsymbol{\theta}$ [4].

2.2. Multi-output Gaussian process

The MOGP intends to approximate the T outputs $\{f_t\}_{1\leq t\leq T}$ simultaneously by considering their correlations, with the aim of outperforming individual modeling. For the convenience of presentation below, we use the isotopic training sets $X_1 = \cdots = X_T = \overline{X} \in R^{n\times d}$, i.e., $n_t = n$ and $\mathbf{x}_{t,i} = \mathbf{x}_i$ for $t = 1, \cdots, T$, though the MOGP can be readily extended to heterotopic training sets.

Similar to the SOGP, the T outputs $\mathbf{f} = \{f_1, \cdots, f_T\}^\mathsf{T}$ are assumed to follow a Gaussian process as

$$\mathbf{f}(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, \mathcal{K}_M(\mathbf{x}, \mathbf{x}')), \tag{9}$$

where the *multi-output covariance* $\mathcal{K}_M(\mathbf{x}, \mathbf{x}') \in R^{T\times T}$ is defined as

$$\mathcal{K}_M(\mathbf{x}, \mathbf{x}') = \begin{bmatrix} k_{11}(\mathbf{x}, \mathbf{x}') & \cdots & k_{1T}(\mathbf{x}, \mathbf{x}') \\ \vdots & \ddots & \vdots \\ k_{T1}(\mathbf{x}, \mathbf{x}') & \cdots & k_{TT}(\mathbf{x}, \mathbf{x}') \end{bmatrix}. \tag{10}$$

The element $k_{tt'}(\mathbf{x}, \mathbf{x}')$ corresponds to the covariance, i.e., the degree of correlation or similarity, between outputs $f_t(\mathbf{x})$ and $f_{t'}(\mathbf{x}')$.

Given the relationship

$$y_t(\mathbf{x}) = f_t(\mathbf{x}) + \epsilon_t, \tag{11}$$

where the *iid* Gaussian noise $\epsilon_t \sim \mathcal{N}(0, \sigma_{s,t}^2)$ is assigned to the t-th output, the likelihood function for the T outputs follows

$$p(\mathbf{y}|\mathbf{f}, \mathbf{x}, \Sigma_s) = \mathcal{N}(\mathbf{f}(\mathbf{x}), \Sigma_s), \tag{12}$$

where $\Sigma_s \in R^{T\times T}$ is a diagonal matrix with the elements $\{\sigma_{s,t}^2\}_{1\leq t\leq T}$. Note that the consideration of ϵ_t can (1) help transfer knowledge across the outputs [34, 35], and (2) capture some output-specific features since it is related to f_t.

Then, given the training set $X = \{X_1, \cdots, X_T\}^\mathsf{T}$ and the output observations $\mathbf{y} = \{\mathbf{y}_1, \cdots, \mathbf{y}_T\}^\mathsf{T}$, the posterior distribution of $\mathbf{f}(\mathbf{x}_*) = \{f_1(\mathbf{x}_*), \cdots, f_T(\mathbf{x}_*)\}^\mathsf{T}$ at a test point $\mathbf{x}_*$ can be analytically derived as

$$\mathbf{f}(\mathbf{x}_*)|X, \mathbf{y}, \mathbf{x}_* \sim \mathcal{N}(\hat{\mathbf{f}}(\mathbf{x}_*), \Sigma_*). \tag{13}$$

The prediction mean and variance are respectively given as

$$\hat{\mathbf{f}}(\mathbf{x}_*) = K_{M*}^\mathsf{T}[K_M(\overline{X}, \overline{X}) + \Sigma_M]^{-1}\mathbf{y}, \tag{14a}$$

$$\Sigma_* = \mathcal{K}_M(\mathbf{x}_*, \mathbf{x}_*) - K_{M*}^\mathsf{T}[K_M(\overline{X}, \overline{X}) + \Sigma_M]^{-1}K_{M*}, \tag{14b}$$

where $K_{M*} = K_M(\overline{X}, \mathbf{x}_*) \in R^{nT\times T}$ has blocks $K_{tt'}(\overline{X}, \mathbf{x}_*) = [k_{tt'}(\mathbf{x}_i, \mathbf{x}_*)]$ for $i = 1, \cdots, n$ and $t, t' = 1, \cdots, T$; $\mathcal{K}_M(\mathbf{x}_*, \mathbf{x}_*) \in R^{T\times T}$ has elements $k_{tt'}(\mathbf{x}_*, \mathbf{x}_*)$

for $t, t' = 1, \cdots, T$; the t-th diagonal element of Σ_* corresponds to $\sigma_t^2(\mathbf{x}_*)$; and $\Sigma_M = \Sigma_s \otimes I_n \in R^{nT \times nT}$ is a diagonal noise matrix; the symmetric and block partitioned matrix $K_M(\overline{X}, \overline{X}) \in R^{nT \times nT}$ is calculated by Eq. (10) as [4]

$$K_M(\overline{X}, \overline{X}) = \begin{bmatrix} K_{11}(\overline{X}, \overline{X}) & \cdots & K_{1T}(\overline{X}, \overline{X}) \\ \vdots & \ddots & \vdots \\ K_{T1}(\overline{X}, \overline{X}) & \cdots & K_{TT}(\overline{X}, \overline{X}) \end{bmatrix}. \tag{15}$$

Similarly, the hyperparameters $\boldsymbol{\theta}_M$, including the parameters in $\{k_{tt'}\}_{1 \leq t,t' \leq T}$ and $\{\sigma_{s,t}^2\}_{1 \leq t \leq T}$, for the T outputs can be inferred by solving problem (7). Note that for the sake of brevity, we use $f_{*t}(\mathbf{x}) \sim \mathcal{GP}(\hat{f}_t(\mathbf{x}), \sigma_t^2(\mathbf{x}))$ to represent the posterior distribution of $f_t(\mathbf{x})$ given the observations throughout the article.

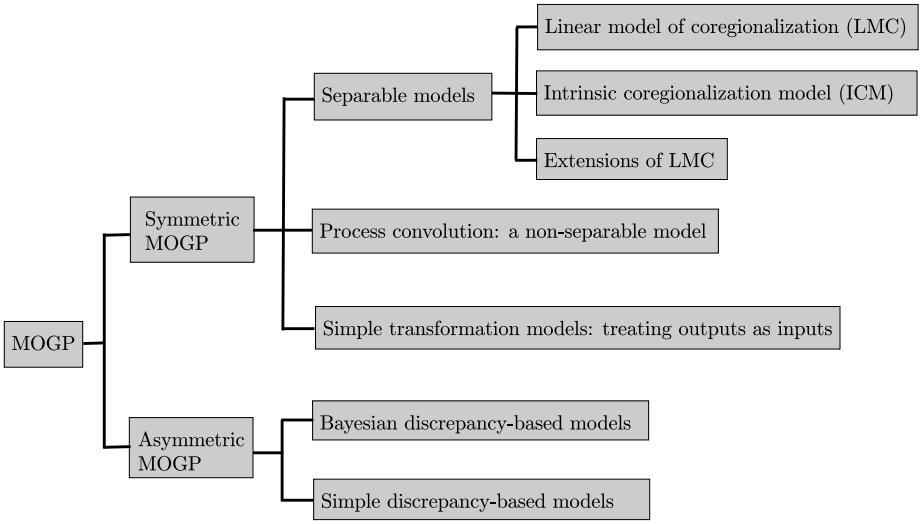

Figure 2: Categories of the existing MOGPs.

It is found that the performance of MOGP depends on the admissible multi-output covariance structure $\mathcal{K}_M(\mathbf{x}, \mathbf{x}')$ that should be capable of (1) building a PSD covariance matrix $K_M(\overline{X}, \overline{X})$ in Eq. (15), and (2) capturing the output correlations and transferring available information across outputs. In the following sections, we review and analyze the characteristics and differences of the existing MOGPs categorized in Fig. 2.

3. Symmetric MOGP

By treating the outputs equally, the key in symmetric MOGP is to develop symmetric covariance functions in order to capture the output correlations for

[4] In general setting, we can write the block in the matrix as $K_{tt'}(X_t, X_{t'}) \in R^{n_t \times n_{t'}}$ for $1 \leq t, t' \leq T$.

sharing useful information across the outputs as much as possible. Here we classify the symmetric MOGPs into three core categories: (1) separable models that treat the inputs and outputs separately; (2) process convolution that adopts a non-separable mixture of inputs and outputs; and (3) simple transformation models that decompose the multi-output process into a series of single-output processes by treating outputs as inputs.

3.1. Separable models

3.1.1. Linear model of coregionalization

The most widely used generative approach for developing admissible multi-output covariance functions was pioneered in the geostatistics community, known as *linear model of coregionalization* (LMC) [36, 37, 38, 39]. The LMC, as shown in Fig. 3, expresses the outputs as a linear combination of Q latent functions as

$$f_t(\mathbf{x}) = \sum_{q=1}^{Q} a_{t,q} u_q(\mathbf{x}), \tag{16}$$

where the latent function $u_q(\mathbf{x})$ is assumed to be a Gaussian process with zero mean and covariance as $\text{cov}[u_q(\mathbf{x}), u_q(\mathbf{x}')] = k_q(\mathbf{x}, \mathbf{x}')$, and $a_{t,q}$ is the coefficient for $u_q(\mathbf{x})$. It is observed that the T outputs in the symmetric LMC model are formulated similarly, and they only differ in the coefficients which measure the output correlations. The LMC model can be expressed in a matrix formulation as

$$\mathbf{f}(\mathbf{x}) = B\mathbf{u}(\mathbf{x}), \tag{17}$$

where $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \cdots, f_T(\mathbf{x})]^\mathsf{T}$, $\mathbf{u}(\mathbf{x}) = [u_1(\mathbf{x}), \cdots, u_Q(\mathbf{x})]^\mathsf{T}$, and $B \in R^{T \times Q}$ is a matrix with the t-th row as $B_{t,:} = [a_{t,1}, \cdots, a_{t,Q}]$.

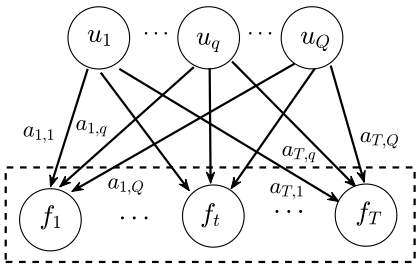

Figure 3: Graphical model of the LMC.

Besides, the LMC model introduces an independent assumption $u_q(\mathbf{x}) \perp u_{q'}(\mathbf{x}')$, i.e., $\text{cov}[u_q(\mathbf{x}), u_{q'}(\mathbf{x}')] = 0$, for $q \neq q'$. Consequently, the cross covari-

ance between two outputs $f_t(\mathbf{x})$ and $f_{t'}(\mathbf{x}')$ can be expressed as

$$\begin{aligned} k_{tt'}(\mathbf{x}, \mathbf{x}') &= \sum_{q=1}^{Q} \sum_{q'=1}^{Q} a_{t,q} a_{t',q'} \mathrm{cov}[u_q(\mathbf{x}), u_{q'}(\mathbf{x}')] \\ &= \sum_{q=1}^{Q} a_{t,q} a_{t',q} k_q(\mathbf{x}, \mathbf{x}'). \end{aligned} \tag{18}$$

Due to the decoupled inputs and outputs in the covariance structure, the LMC is known as a *separable* model [40]. Besides, it has been pointed out that the linear combination of several covariance functions still results in a valid covariance function, i.e., $k_{tt'}(\mathbf{x}, \mathbf{x}')$ in Eq. (18) is admissible [4]. Next, the multi-output covariance $\mathcal{K}_M(\mathbf{x}, \mathbf{x}')$ in Eq. (10) can be expressed as

$$\mathcal{K}_M(\mathbf{x}, \mathbf{x}') = \sum_{q=1}^{Q} A_q k_q(\mathbf{x}, \mathbf{x}'), \tag{19}$$

where $A_q \in R^{T \times T}$ is a symmetric and PSD correlation matrix (also called coregionalization matrix) wherein the element $A^q_{tt'} = a_{t,q} a_{t',q}$.

Particularly, if $Q = 1$ we derive the *intrinsic coregionalization model* (ICM) [41], with the multi-output covariance simplified as

$$\mathcal{K}_M(\mathbf{x}, \mathbf{x}') = A k(\mathbf{x}, \mathbf{x}'), \tag{20}$$

where $A \in R^{T \times T}$ has the element $A_{tt'} = a_t a_{t'}$ that represents the correlation between $f_t(\mathbf{x})$ and $f_{t'}(\mathbf{x}')$. Given the training points $X_1 = \cdots = X_T = \overline{X}$, we have the covariance matrix for the ICM model as

$$K_M(\overline{X}, \overline{X}) = A \otimes K(\overline{X}, \overline{X}). \tag{21}$$

Compared to the LMC, the ICM is more restrictive since it uses only a single latent process to capture the possible variability across outputs. But due to the Kronecker product, the computational complexity of the ICM can be greatly reduced [34].

3.1.2. Considerations of implementing LMC/ICM

In order to implement the LMC/ICM for multi-output modeling, we need to specify the latent covariance functions $\{k_q\}_{1 \le q \le Q}$ and learn the correlation matrices $\{A_q\}_{1 \le q \le Q}$. As for the Q latent covariance functions, all of them can simply use the SE function in Eq. (2) but own different hyperparameters $\{\boldsymbol{\theta}_q\}_{1 \le q \le Q}$; or furthermore, they can choose completely different function types. As for the number Q of covariance functions, it has been pointed out that a large value of Q produces an expressive and flexible model. For example, if the outputs have different characteristics, e.g., different length scales, the $Q > 1$ covariances are capable of describing such variability [40]. The Q value is unrestricted, and in practice some researchers have recommended, e.g., $Q = 2$ latent processes

[42], or $Q = T$ latent processes [43]. The further increase of Q seems to yield little improvements, while significantly increasing the computational demands since we need to infer many hyperparameters in the Q latent processes.

In terms of learning the correlation matrix, the simplest way is to have $A_q = I$, which indicates that the outputs are uncorrelated. Consequently, the matrix $K_M(\overline{X}, \overline{X})$ is block diagonal. In this case, the output correlations are buried in k_q by sharing the hyperparameters $\boldsymbol{\theta}_q$ across all the outputs [44]. There are some other parameterization choices of A_q, e.g., $A_q = \mathbf{1}$ and $A_q = \mathbf{1} + \alpha I$ [45]. To improve flexibility, Osborne et al. [46] employed the ICM model and parameterized the correlation matrix as $A = \text{diag}(\mathbf{e})S^{\mathsf{T}}S\text{diag}(\mathbf{e})$, where $\mathbf{e}$ corresponds to the length scale of each output and S is an upper triangular matrix describing the particular spherical coordinates of points. Besides, the semi-parametric latent factor model (SLFM) [47] defines the correlation matrix as $A_q = \boldsymbol{a}_q \boldsymbol{a}_q^{\mathsf{T}}$ where $\boldsymbol{a}_q = \{a_{t,q}\}_{1 \leq t \leq T}$. In this case, A_q has a rank of one, and is equivalent to the free-form parameterization in what follows for $P = 1$.

Bonilla et al. [34] introduce a general "free-form" strategy to parameterize A_q based on the Cholesky decomposition $A_q = LL^{\mathsf{T}}$, which guarantees the positive semi-definiteness of A_q. The lower triangular matrix L is parameterized as

$$L = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ a_2 & a_3 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ a_{w-T+1} & a_{w-T+2} & \cdots & a_w \end{bmatrix}, \tag{22}$$

where $w = T(T+1)/2$ is the number of correlation parameters. It has been pointed out that the free-form parameterization allows the elements of A_q to scale freely for each pair of outputs, thus enhancing the ability to well estimate the output correlations [13]. However, a full rank A_q has $T(T+1)/2$ correlation parameters, leading to a high dimensional inference problem with the increase of outputs. To mitigate this issue, an incomplete-Cholesky decomposition is used in [34] to build a rank-P approximation of A_q as

$$A_q \approx \widetilde{A}_q = \widetilde{L}\widetilde{L}^{\mathsf{T}}, \tag{23}$$

where $\widetilde{L}$ is a $T \times P$ ($P \leq T$) matrix, and consequently, the rank of A_q now is P. With the increase of P, Eq. (23) is enabled to estimate the output correlations more accurately and flexibly, but requiring more computational cost due to the increasing number of correlation parameters. The ICM model with the correlation matrix A parameterized by Eq. (23) is referred to as multi-task Gaussian process (MTGP) [34]. For problems with several outputs, we prefer using a full rank A_q; while for problems with many outputs, e.g., a dataset with 139 outputs in [34], the P-rank parameterization is required for alleviating computational budget.

3.1.3. Extensions of the LMC

In the LMC framework, the latent function $u_q(\mathbf{x})$ shares the same hyperparameters $\boldsymbol{\theta}_q$ for all the outputs. As a result, these Q shared processes $\{u_q\}_{1 \leq q \leq Q}$

help transfer the common information across outputs. However, except the correlated common features, the outputs themselves may have some unique features that have not yet been considered in the LMC framework.

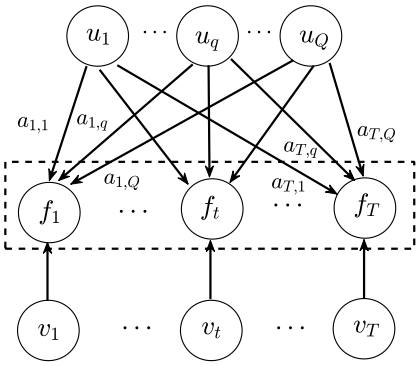

Figure 4: Graphical model of the CoMOGP.

Therefore, as shown in Fig. 4, Nguyen et al. [42] proposed a collaborative MOGP (CoMOGP) based on the *common-specific decomposition* as

$$f_t(\mathbf{x}) = \sum_{q=1}^{Q} a_{t,q} u_q(\mathbf{x}) + v_t(\mathbf{x}), \tag{24}$$

where the first Q shared processes in the right-hand side correspond to the common features of the outputs, whereas the output-specific process $v_t(\mathbf{x})$ corresponds to the specific features of $f_t(\mathbf{x})$ itself. Similar to the role of $Q > 1$ in LMC, the output-specific process also has the ability to explicitly capture the variability of outputs. Besides, this "explaining away" term may help reduce negative transfer across outputs [48].

Based on the independent assumptions $u_q(\mathbf{x}) \perp v_t(\mathbf{x}')$ and $u_q(\mathbf{x}) \perp u_{q'}(\mathbf{x}')$ for $1 \leq q \neq q' \leq Q$ and $1 \leq t \leq T$, the covariance function between $f_t(\mathbf{x})$ and $f_{t'}(\mathbf{x}')$ is given as

$$k_{tt'}(\mathbf{x}, \mathbf{x}') = \begin{cases} \sum_{q=1}^{Q} a_{t,q}^2 k_q(\mathbf{x}, \mathbf{x}') + k_t(\mathbf{x}, \mathbf{x}'), & t = t', \\ \sum_{q=1}^{Q} a_{t,q} a_{t',q} k_q(\mathbf{x}, \mathbf{x}'), & t \neq t'. \end{cases} \tag{25}$$

Consequently, the multi-output covariance is written as $\mathcal{K}_M(\mathbf{x}, \mathbf{x}') = [k_{tt'}(\mathbf{x}, \mathbf{x}')]_{1 \leq t, t' \leq T}$. The common-specific decomposition idea has also been explored in [19], known as *common component model* (CCM). A recent work [49] further decomposes the specific process $v_t(\mathbf{x})$ into a weighted-sum formulation in order to handle diverse data structures in multi-output modeling.

There are some other extensions of typical LMC, for example, the non-stationary MOGP by using a spatially varying correlation matrix [50, 51], the combination of the LMC and the Bayesian treed Gaussian process [52, 53], and the self-measuring MTGP that uses the information of outputs to construct informative correlation matrix [54, 55].

3.2. Process convolution: a non-separable model

As has been introduced before, the LMC model uses a linear combination of several independent latent processes to represent the correlated outputs. The performance of this separable model, especially the ICM model, may be limited in some scenarios where for example one output is a blurred version of the other [56]. This is because the model shares the same hyperparameters for a latent process across the outputs, see Eq. (16). Here we introduce another way beyond the separable model, called *process convolution* (CONV). It is a *non-separable* generative model that can build valid covariance functions for multi-output modeling by convolving a base process with a smoothing kernel. Different from the LMC, the flexible CONV allows to mimic each output using individual hyperparameters. Besides, it has been pointed out that the convolved process is ensured to be a Gaussian process if the base process is a Gaussian process, which makes it analytically tractable.

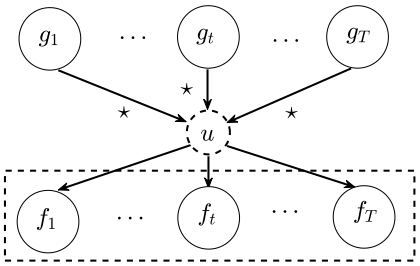

Figure 5: Graphical model of the CONV, where the symbol $\star$ represents a convolution operation.

As shown in Fig. 5, in the convolved multi-output modeling framework, the output $f_t(\mathbf{x})$ can be expressed by convolving a smoothing and output-dependent kernel $g_t(.)$ with a common base process $u(.)$ as [17]

$$f_t(\mathbf{x}) = \int_{-\infty}^{\infty} g_t(\mathbf{x} - \mathbf{z}) u(\mathbf{z}) d\mathbf{z}. \tag{26}$$

This expression captures the shared and output-specific features of the outputs by a mixture of common and specific latent processes [43]. With the independence assumption $g_t \perp g_{t'}$, the cross covariance between $f_t(\mathbf{x})$ and $f_{t'}(\mathbf{x}')$ is

$$k_{tt'}(\mathbf{x}, \mathbf{x}') = \int_{-\infty}^{\infty} g_t(\mathbf{x} - \mathbf{z}) g_{t'}(\mathbf{x}' - \mathbf{z}') k(\mathbf{z}, \mathbf{z}') d\mathbf{z} d\mathbf{z}'. \tag{27}$$

Alvarez et al. [56] pointed out that if we take the smoothing kernel to be the Dirac delta function, i.e., $g_t(\mathbf{x}-\mathbf{z}) = a_t\delta(\mathbf{x}-\mathbf{z})$, Eq. (27) degenerates to the LMC model in Eq. (16) with $Q = 1$. That is, compared to the static mixture in the LMC, the CONV can be regarded as a dynamic version of the LMC due to the smoothing kernel. Note that for defining more complex covariance functions, Eq. (27) can be extended with $Q > 1$, i.e., allowing multiple convolutions, see [56]. Moreover, similar to the common-specific decomposition idea, Boyle et al. [57] developed a dependent GP model using multiple convolutions corresponding to different features of the outputs.

In practice, we usually assume that the base process $u(\mathbf{z})$ is a white Gaussian noise [17]. Consequently, Eq. (27) can be simplified as

$$k_{tt'}(\mathbf{x},\mathbf{x}') = \int_{-\infty}^{\infty} g_t(\mathbf{x}-\mathbf{z})g_{t'}(\mathbf{x}'-\mathbf{z})d\mathbf{z}. \tag{28}$$

By using fourier analysis and SE function for the smoothing kernel, the covariance can be analytically derived as [58]

$$\begin{aligned} k_{tt'}(\mathbf{x},\mathbf{x}';\sigma_{f,t},P_t,\sigma_{f,t'},P_{t'}) = 2^{n/2}\sigma_{f,t}\sigma_{f,t'}\frac{|P_t|^{1/4}P_{t'}|^{1/4}}{\sqrt{|P_t+P_{t'}|}} \\ \times \exp\left(-(\mathbf{x}-\mathbf{x}')^{\mathsf{T}}(P_t+P_{t'})^{-1}(\mathbf{x}-\mathbf{x}')\right), \end{aligned} \tag{29}$$

where $\sigma_{f,t}$ and $\sigma_{f,t'}$ are the signal variances for $f_t(\mathbf{x})$ and $f_{t'}(\mathbf{x}')$, respectively; the diagonal matrix $P_t \in R^{d\times d}$ contains the length scales along each input direction for output $f_t(\mathbf{x})$, and similarly, $P_{t'}$ has the length scales for $f_{t'}(\mathbf{x}')$. We can see that Eq. (29) is equivalent to the original SE covariance function in Eq. (2) when $t = t'$, thus allowing to preserve the individual characteristics of each output in the multi-output modeling framework. Note that the base process can be extended as a general Gaussian process rather than a white noise [59, 60]. For some recent variants of the CONV models, please refer to [61, 62].

It is found that the hyperparameters in Eq. (29) are related to each output, leaving no free parameters to explicitly describe the output correlations. To complete and enhance the multi-output modeling process, we need to specify the parameters to capture the output correlations, e.g., by the free-form strategy in Eq. (23). As a result, the multi-output covariance for the CONV model can be expressed as $\mathcal{K}_M(\mathbf{x},\mathbf{x}') = [a_{tt'}k_{tt'}(\mathbf{x},\mathbf{x}')]_{1\le t,t'\le T}$. Wagle et al. [62] developed a similar model, called forward adaptive transfer GP (FAT-GP), for two-output cases by using the covariance in Eq. (29) and a single output similarity factor λ. There are also some other ways, e.g., space partition [63], to enable the freedom in capturing output correlations.

Last, we have known that the CONV model is able to take into account the output-specific features by using an output-dependent convolution process, see Eq. (26). As has been pointed out before, we can improve the ability of the LMC to capture output-specific features by increasing the Q value, i.e., adopting more latent processes with different characteristics. Another way to achieve this is

to employ a framework like the common-specific decomposition in Eq. (24) to explicitly account for the output-specific features. One advantage of the CONV is that it offers a simpler model to describe the data [56].

3.3. Simple transformation models: treating outputs as inputs

The above Bayesian MOGPs employ an integrated modeling process wherein all the information of the outputs are fused in a multi-output covariance matrix $K_M(\overline{X}, \overline{X})$, and the hyperparameters are inferred jointly. Recently, some simple multi-output models that employ a kind of *decomposed* modeling process have been proposed. These models are easy to implement and can be applied to any surrogate model. Borchani et al. [30] denoted them as problem transformation methods, because they attempt to transform the multi-output modeling process into a series of successive single-output modeling processes by treating outputs as inputs. In what follows, we introduce two simple multi-output modeling approaches [64], *stacked single-target* (SST) and *ensemble of regressor chains* (ERC), that are motivated by some successful multi-label classification approaches.

3.3.1. Stacked single-target

The SST is developed from the *stacked generalization* idea [65] in multi-label classification. The stacking strategy has also been used for multi-output sampling [66]. The key of SST is to use the predicted outputs as inputs to correct the predictions.

The SST training process has two stages. In the first stage, we use the training data $\mathcal{D}_t = \{X_t, \mathbf{y}_t\}$ to build the first-stage model $f_{*t}(\mathbf{x})$ for each of the T outputs. In the second stage, we augment a point $\mathbf{x}$ by involving the predictions as $\mathbf{x}' = [\mathbf{x}, \hat{f}_1(\mathbf{x}), \cdots, \hat{f}_T(\mathbf{x})]$. Thereafter, we use the transformed training data $\mathcal{D}'_t = \{X'_t, \mathbf{y}_t\}$ to learn a new second-stage surrogate model $f'_{*t}(\mathbf{x}')$ for each output.

The prediction process at a test point $\mathbf{x}_*$ also has two stages. In the first stage we use the first-stage models $\{f_{*t}(\mathbf{x})\}_{1\leq t\leq T}$ to have predictions at $\mathbf{x}_*$ in order to transform it as $\mathbf{x}'_* = [\mathbf{x}_*, \hat{f}_1(\mathbf{x}_*), \cdots, \hat{f}_T(\mathbf{x}_*)]$. Then, we use the second-stage models $\{f'_{*t}(\mathbf{x}')\}_{1\leq t\leq T}$ to produce the final predictions at $\mathbf{x}'_*$ for the outputs.

3.3.2. Ensemble of regressor chains

The ERC is derived from the idea of multi-label classifier chains that chains single-output models [67]. It begins by selecting a random chain of the outputs and then builds surrogates for the ordered outputs successively. Given an randomly permuted chain set C where the integer $1 \leq C_t \leq T$ represents the index of an output, the training process of ERC is described as below. First, the training data $\mathcal{D}_{C_1} = \{X_{C_1}, \mathbf{y}_{C_1}\}$ is used to fit a model $f_{*C_1}(\mathbf{x})$ for $f_{C_1}(\mathbf{x})$. Then, the subsequent surrogate model $f_{*C_t}(\mathbf{x})$ for $f_{C_t}(\mathbf{x})$ is fitted to a transformed training data $\mathcal{D}'_{C_t} = \{X'_{C_t}, \mathbf{y}_{C_t}\}$ where $X'_{C_t} = [X'_{C_t}, \mathbf{y}_{C_1}, \cdots, \mathbf{y}_{C_{t-1}}]$.

The prediction process at a test point $\mathbf{x}_*$ is similar. We first use the model $f_{*C_1}(\mathbf{x})$ to have a prediction $\hat{f}_{C_1}(\mathbf{x}_*)$ for the first output. Then, we transform

the point $\mathbf{x}_*$ as $\mathbf{x}'_* = [\mathbf{x}_*, \hat{f}_{C_1}(\mathbf{x}_*)]$, and use $f_{*C_2}(\mathbf{x})$ to predict for the second output at $\mathbf{x}'_*$. We can have the predictions for all the outputs at $\mathbf{x}_*$ by repeating the above process.

The main disadvantage of ERC is that it is sensitive to the selection of chains order C [67]. Alternatively, an ensemble strategy can be employed where we generate s ($s \leq T!$) chains and the final predictions come from averaging the results of s chains.

3.3.3. Discussions

The main advantage of the simple models is the ease of implementation with existing surrogates. Besides, some theoretical insights [68, 69, 70] pointed out that the consideration of additional features to transform the training data helps SST and ERC reduce the model bias at the cost of increasing the model variance. The main drawback is that the SST and ERC are hard to understand and interpret for multi-output cases. That is, they have no clear mechanism to describe the output correlations, which may affect the prediction quality. Besides, the performance of these simple strategies are affected by the base surrogate models. Since the GP is a strong surrogate model, it may be difficult to gain improvements over individual modeling using the SST and ERC methods.

4. Asymmetric MOGP

This article particularly focuses on the hierarchical multi-fidelity scenario where the T outputs represent the simulators with different levels of fidelity for the same task. These outputs $\{f_t\}_{1\leq t\leq T}$ are sorted by increasing fidelity, i.e., f_T has the highest fidelity and f_1 has the lowest fidelity. Besides, they have a hierarchical training sets as $n_1 > \cdots > n_T$. Hence, the asymmetric MOGP devotes to transferring information from the inexpensive $T-1$ LF outputs for enhancing the modeling of the expensive HF output.

The LF outputs in the multi-fidelity scenario usually come from simplified analysis models by using, e.g., (a) the coarse finite element meshes; (b) the relaxed boundary or convergence conditions; and (c) the Euler governing equations instead of the Navier-Stokes viscous equations. In practice, we cannot afford extensive HF simulations at many training points. Hence, the multi-fidelity modeling utilizes the correlated yet inexpensive LF information to enhance the expensive HF modeling.

An asymmetric dependency structure is required for effective asymmetric MOGPs. In the multi-fidelity scenario, the discrepancy-based MOGPs have gained popularity. We classify them into two categories, including Bayesian discrepancy-based models and simple discrepancy-based models.

4.1. Bayesian discrepancy-based models

Similar to Eq. (17), the discrepancy-based asymmetric model can be described in the separable LMC-like framework as

$$\mathbf{f}(\mathbf{x}) = B_{\mathsf{L}}\mathbf{u}(\mathbf{x}). \tag{30}$$

To have a hierarchical and ordered structure for the asymmetric scenario, $B_{\mathsf{L}} \in R^{T\times T}$ now is a lower triangular matrix, and $\mathbf{u}(\mathbf{x})$ contains T latent processes. Particularly, based on a Markov property (see Eq. (48)), Kennedy et al. [71] presented an autoregressive model, a variant of Eq. (30), for different levels of fidelity. This model is an extension to the Co-Kriging (CoKG) model that was developed in the geostatistics community as a multi-variate Kriging [72], and can be expressed in a recursive form as

$$f_t(\mathbf{x}) = a_{t-1} f_{t-1}(\mathbf{x}) + \delta_t(\mathbf{x}), \quad 2 \le t \le T, \tag{31}$$

where a_{t-1} represents the correlation factor between adjacent fidelity levels $f_t(\mathbf{x})$ and $f_{t-1}(\mathbf{x})$, and $\delta_t(\mathbf{x})$ represents the discrepancy between $f_t(\mathbf{x})$ and $f_{t-1}(\mathbf{x})$. This model follows the independent assumption that $f_{t-1}(\mathbf{x}) \perp \delta_t(\mathbf{x})$. Compared to the symmetric model in Eq. (16), the model, as shown in Fig. 6, uses a recursive asymmetric structure to achieve the information transfer from the LF outputs to the HF output.

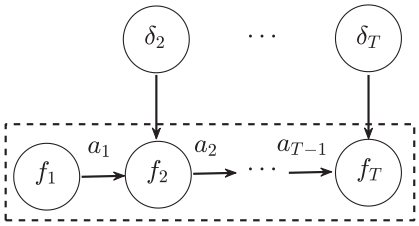

Figure 6: Graphical model of the CoKG.

For the CoKG model, the cross and self covariances can be formulated as

$$k_{t(t-1)}(\mathbf{x}, \mathbf{x}') = a_{t-1} k_{t-1}(\mathbf{x}, \mathbf{x}'), \tag{32a}$$

$$k_t(\mathbf{x}, \mathbf{x}') = a_{t-1}^2 k_{t-1}(\mathbf{x}, \mathbf{x}') + k_{\delta_t}(\mathbf{x}, \mathbf{x}'), \tag{32b}$$

where $k_{t-1}(\mathbf{x}, \mathbf{x}')$ is the covariance of $f_{t-1}(\mathbf{x})$, and $k_{\delta_t}(\mathbf{x}, \mathbf{x}')$ is the covariance of $\delta_t(\mathbf{x})$. More deeply, based on the recursive formulation in (31), we have

$$k_{tt'}(\mathbf{x}, \mathbf{x}') = \left(\prod_{i=t'}^{t-1} a_i\right) k_{t'}(\mathbf{x}, \mathbf{x}'), \quad \forall t > t', \tag{33a}$$

$$k_t(\mathbf{x}, \mathbf{x}') = \left(\prod_{i=1}^{t-1} a_i^2\right) k_1(\mathbf{x}, \mathbf{x}') + \sum_{j=2}^{t-1} \left(\prod_{i=j}^{t-1} a_i^2\right) k_{\delta_j}(\mathbf{x}, \mathbf{x}') + k_{\delta_t}(\mathbf{x}, \mathbf{x}'). \tag{33b}$$

Consequently, the two equations are used to form the multi-output covariance $\mathcal{K}_M(\mathbf{x}, \mathbf{x}') = [k_{tt'}]_{1\le t,t'\le T}$. Note that Qian et al. [73] and Leen et al. [48] also provided an equivalent modeling framework in different views.

It is found that the CoKG model is similar to the CoMOGP model: both of them employ the "explaining away" term to represent the specific features,

which eases the multi-output modeling process. Particularly, the CoKG considers all the specific features in an asymmetric expression, which enables the asymmetric transfer from the LF outputs to the HF output.

Several improvements regarding the CoKG model have been developed recently. To speed up the model efficiency, Le Gratiet et al. [74, 75] derived a recursive formulation that is capable of building a T-level CoKG model through building T independent Kriging models successively. Burnaev et al. [76] and Zaytsev et al. [28] enabled the CoKG to handle a large training size via sparse approximation. A recent interesting work by Perdikaris et al. [77] extended CoKG to handle cases with up to 10^5 inputs and 10^5 training points through data-driven dimensionality reduction techniques and a graph-theoretic approach.

In order to improve the prediction quality, some works [27, 78] attempted to utilize the gradient information in the context of CoKG. Given a limited computational budget, Zaytsev et al. [79] derived a formula to determine the optimal shares of variable fidelity points by minimizing the maximal interpolation errors of the CoKG model and taking into account the computing ratios between fidelity levels. Besides, instead of using a constant correlation factor, Han et al. [27] parametrized it via a polynomial form and organized the model as

$$f_t(\mathbf{x}) = \boldsymbol{p}^{\mathsf{T}}(\mathbf{x})\boldsymbol{a}_{t-1} f_{t-1}(\mathbf{x}) + \delta_t(\mathbf{x}), \tag{34}$$

where $\boldsymbol{p}(\mathbf{x}) = [1, p_1(\mathbf{x}), \cdots, p_q(\mathbf{x})]^{\mathsf{T}}$ contains $1+q$ basis functions, and $\boldsymbol{a}_{t-1} = [a_0, \cdots, a_q]^{\mathsf{T}}$ is the corresponding correlation factors. This flexible formulation enables CoKG to handle multi-fidelity cases with nonlinear correlations. Furthermore, Perdikaris et al. [80] generalized the CoKG model as

$$f_t(\mathbf{x}) = z_{t-1}(f_{t-1}(\mathbf{x})) + \delta_t(\mathbf{x}), \tag{35}$$

where $z_{t-1}(\mathbf{x})$ is a function that maps $f_{t-1}(\mathbf{x})$ to $f_t(\mathbf{x})$, and can be assigned with a GP prior. The nonlinear mapping of a Gaussian distribution $z_{t-1}(f_{t-1}(\mathbf{x}))$, however, is intractable in practice. To keep the analytical tractability of the Bayesian model, the authors replaced the GP prior of $f_{t-1}(\mathbf{x})$ with the posterior distribution $f_{*t-1}(\mathbf{x})$ from the previous fidelity level. Thereafter, a structured covariance function is introduced as

$$k_t(\mathbf{x}, \mathbf{x}') = k_{z_{t-1}}(\mathbf{x}, \mathbf{x}') k_{t-1}(f_{*t-1}(\mathbf{x}), f_{*t-1}(\mathbf{x}')) + k_{\delta_t}(\mathbf{x}, \mathbf{x}'). \tag{36}$$

Note that in this nonlinear CoKG (nlCoKG) model, the posterior distribution $p(f_t(\mathbf{x}_*)|X_t, \mathbf{y}_t, \mathbf{x}_*)$ is no longer Gaussian, since the point is transformed to $[\mathbf{x}_*, f_{*t-1}(\mathbf{x}_*)]$ that contains uncertain inputs. To obtain the posterior distribution, we can integrate out $f_{*t-1}(\mathbf{x}_*)$ as

$$p(f_{*t}(\mathbf{x}_*)) = \int p(f_t(\mathbf{x}_*, f_{*t-1}(\mathbf{x}_*))) p(f_{*t-1}(\mathbf{x}_*)) d\mathbf{x}_*. \tag{37}$$

This model has been showed to be capable of handling multi-fidelity data with nonlinear, non-functional and space-dependent output correlations. Another improvement is achieved by the calibration strategy that is a means to tune

physical parameters in order to obtain the best agreement between the predictions and the actual values. Kennedy et al. [81] presented a calibration CoKG model as

$$f_t(\mathbf{x}) = a_{t-1} f_{t-1}(\mathbf{x}; \boldsymbol{\theta}_c) + \delta_t(\mathbf{x}), \tag{38}$$

where $\boldsymbol{\theta}_c$ is the calibration parameters that should be tuned in order to achieve the best agreement between $f_t(\mathbf{x})$ and $f_{t-1}(\mathbf{x})$. The calibration framework brings about great flexibility at the cost of estimating many hyperparameters.

4.2. Simple discrepancy-based models

Instead of building the models in a Bayesian view, there are some simple multi-fidelity models, e.g., the multiplicative model $f_t(\mathbf{x}) = a_{t-1} f_{t-1}(\mathbf{x})$ [82, 83] and the additive model $f_t(\mathbf{x}) = f_{t-1}(\mathbf{x}) + \delta_t(\mathbf{x})$ [84]. The simple discrepancy-based model (SDM) $f_t(\mathbf{x}) = a_{t-1} f_{t-1}(\mathbf{x}) + \delta_t(\mathbf{x})$ further combines the multiplicative model and the additive model to yield better results [85, 29]. The SDM provides a decomposed modeling process by building the models for the outputs individually and estimating the correlations separately, which makes it easy to implement and be extended to existing surrogates, e.g., radial basis functions (RBF) [86, 87, 88].

The SDM usually decomposes the modeling process into four steps: (1) fit the low fidelity model $f_{*t-1}(\mathbf{x})$; (2) estimate the correlation factor a_{t-1} using an error criterion; (3) fit the discrepancy model $\delta_{*t}(\mathbf{x})$ to the training data $\{X_t, \mathbf{y}_t - a_{t-1}\hat{f}_{t-1}(X_t)\}$; and finally (4) build the surrogate model as $f_{*t}(\mathbf{x}) = a_{t-1} f_{*t-1}(\mathbf{x}) + \delta_{*t}(\mathbf{x})$. In step (1), if $t = 2$, we fit $f_{*1}(\mathbf{x})$ to the training data $\{X_1, \mathbf{y}_1\}$; otherwise, we repeat the above steps to obtain $f_{*t-1}(\mathbf{x}) = a_{t-2} f_{*t-2}(\mathbf{x}) + \delta_{*t-1}(\mathbf{x})$. In step (2), the factor a_{t-1} is often optimized as [29]

$$a_{t-1} = \arg\min_{a} \sum_{\mathbf{x} \in X_t} |y_t(\mathbf{x}) - a\hat{f}_{t-1}(\mathbf{x})|. \tag{39}$$

In our testing, the criterion (39) often cannot provide a good estimation of a_{t-1}. We present here a new criterion to obtain a_{t-1} when using the GP to approximate the outputs. It is found that the prediction variance of $f_{*t}(\mathbf{x})$ is expressed as

$$\sigma_t^2(\mathbf{x}) = a_{t-1}^2 \sigma_{t-1}^2(\mathbf{x}) + \sigma_{\delta_t}^2(\mathbf{x}; a_{t-1}), \tag{40}$$

where $\sigma_{t-1}^2(\mathbf{x})$ is the prediction variance of $f_{*t-1}(\mathbf{x})$, and $\sigma_{\delta_t}^2(\mathbf{x}; a_{t-1})$ is the prediction variance of $\delta_{*t}(\mathbf{x})$. Note that $\sigma_{\delta_t}^2$ depends on the factor a_{t-1} because $\mathbf{y}_{\delta_t} = \mathbf{y}_t - a_{t-1}\hat{f}_{t-1}(X_t)$. According to the *bias-variance* decomposition [89], the generalization error e_t of $f_{*t}(\mathbf{x})$ is

$$e_t = \int (y_t(\mathbf{x}) - \hat{f}_t(\mathbf{x}))^2 + \sigma_t^2(\mathbf{x}) d\mathbf{x}. \tag{41}$$

The bias term $(y_t(\mathbf{x}) - \hat{f}_t(\mathbf{x}))^2$ is usually unknown because of the unaccessible exact observation $y_t(\mathbf{x})$ at an unobserved point $\mathbf{x}$; however, the variance term $\sigma_t^2(\mathbf{x})$ is available for the GP model. Hence, in order to reduce the generalization

error of the model, we can decrease the integrated mean square error (IMSE) $\int_{\mathbf{x}} \sigma_t^2(\mathbf{x})d\mathbf{x}$. Along this line, the correlation factor can be optimized as

$$a_{t-1} = \arg\min_{a} \int a^2\sigma_{t-1}^2(\mathbf{x}) + \sigma_{\delta_t}^2(\mathbf{x}; a)d\mathbf{x}. \tag{42}$$

Eq. (42) attempts to obtain an optimal a_{t-1} to minimize the overall uncertainty of $f_{*t}(\mathbf{x})$, thus minimizing the generalization error e_t.

5. Inference and computational considerations

In order to implement the above reviewed MOGPs, we need to infer the hyperparameters and the correlation parameters (also treated as hyperparameters if possible). Similar to the SOGP, the hyperparameters can be inferred by maximizing the marginal likelihood $p(\mathbf{y}|X, \boldsymbol{\theta}_M)$, i.e., solving an auxiliary min-NLML problem as (7). The partial derivatives of the NLML w.r.t. the hyperparameters $\boldsymbol{\theta}_M$ are analytically derived as [4]

$$\begin{aligned}\frac{\partial \text{NLML}}{\partial \theta_M^j} &= -\frac{1}{2}\mathbf{y}^T K_y^{-1}\frac{\partial K_y}{\partial \theta_M^j}K_y^{-1}\mathbf{y} + \frac{1}{2}\text{tr}\left(K_y^{-1}\frac{\partial K_y}{\partial \theta_M^j}\right) \\ &= \frac{1}{2}\text{tr}\left((K_y^{-1} - \boldsymbol{\alpha}\boldsymbol{\alpha}^T)\frac{\partial K_y}{\partial \theta_M^j}\right),\end{aligned} \tag{43}$$

where $K_y = K_M(\overline{X}, \overline{X}) + \Sigma_M$ and $\boldsymbol{\alpha} = [K_M(\overline{X}, \overline{X}) + \Sigma_M]^{-1}\mathbf{y}$. With the available derivative information, we employ the efficient gradient descent algorithm to solve problem (7).

It is found that the computational complexity of problem (7) depends on the calculation of $K_M^{-1}(\overline{X}, \overline{X})$ and the dimensionality (i.e., the number of hyperparameters in $\boldsymbol{\theta}_M$). Suppose that this article adopts the SE covariance function k_{SE} in Eq. (2) that contains $d+1$ hyperparameters for all the latent processes in the MOGPs throughout. Table 1 lists the computational complexity of calculating K_M^{-1} in each of ten representative MOGPs in the second column; besides, the last three columns provide the number of hyperparameters in each MOGP, including the n_c correlation parameters, the n_{SE} covariance parameters and the n_{noise} noise parameters.

In terms of the computational complexity of K_M^{-1}, it represents (1) the type of modeling process (integrated or decomposed), (2) the type of information transfer (bidirectional or unidirectional) and (3) the type of hyperparameter inference (joint or separate). It is found that the first four Bayesian symmetric MOGPs (MTGP, SLFM, CoMOGP and CONV) employ an integrated multi-output covariance matrix $K_M(\overline{X}, \overline{X})$ in Eq. (15) to fuse the information of all the outputs. Hence, they have to calculate the inverse of an $N \times N$ ($N = \sum_{t=1}^{T} n_t$) matrix, leading to the computational complexity of $\mathcal{O}(N^3)$. The integrated modeling process means that the information is transferred bidirectionally between the outputs, which is beneficial for the improvement of all the outputs.

Table 1: The characteristics of ten representative MOGPs, including the computational complexity of K_M^{-1} and the number of hyperparameters to be inferred.

Model	K_M^{-1}	n_c	n_{SE}	n_{noise}
MTGP [34]	$\mathcal{O}(N^3)$	$T(T+1)/2$	$d+1$	T
SLFM [47]	$\mathcal{O}(N^3)$	$QT(T+1)/2$	$Q(d+1)$	T
CoMOGP [42]	$\mathcal{O}(N^3)$	$QT(T+1)/2$	$(T+Q)(d+1)$	T
CONV [58]	$\mathcal{O}(N^3)$	$T(T+1)/2$	$T(d+1)$	T
SST [64]	$\mathcal{O}((n+T)^3)$	N/A	$T(d+1)+T(d+T+1)$	$2T$
ERC [64]	$\mathcal{O}((n+T)^3)$	N/A	$s[T(d+1)+T(T-1)/2]$	sT
CoKG$_a$ [71]	$\mathcal{O}(N^3)$	$T-1$	$T(d+1)$	T
CoKG$_b$ [15]	$\mathcal{O}(n^3)$	$T-1$	$T(d+1)$	T
nlCoKG [80]	$\mathcal{O}(n^3)$	$(T-1)(d+1)$	$2(T-1)+T(d+1)$	T
SDM	$\mathcal{O}(n^3)$	$T-1$	$T(d+1)$	T

Also, an inevitable situation arises that the hyperparameters $\boldsymbol{\theta}_M$, including the correlation parameters, the covariance parameters and the noise parameters, for all the outputs should be inferred simultaneously, which makes the min-NLML problem (7) a non-trivial high-dimensional optimization task. Contrarily to the four Bayesian symmetric MOGPs, the two transformation models SST and ERC, which treat outputs as inputs, need to invert a covariance matrix with the size up to $(n+T)\times(n+T)$. But due to the decomposed modeling process, the hyperparameters for each output can be inferred separately.

As for the four asymmetric MOGPs (CoKG$_a$, CoKG$_b$, nlCoKG and SDM), three of them are allowed to decompose the multi-output model into a series of sub-models. Consequently, given $n_1 = \cdots = n_T = n$, they only need to invert an $n \times n$ covariance matrix $K(\overline{X}, \overline{X})$, reducing the computational complexity to $\mathcal{O}(n^3)$. Besides, the hierarchical and decomposed modeling process lets them unidirectionally transfer information from the LF outputs to the HF output. At the same time, they can infer the hyperparameters for the outputs separately. Note that the decomposed modeling process is supported by the fact that the inexpensive LF outputs can afford more training points than the HF output in the multi-fidelity scenario.

It is worth noting that for the CoKG model [71], the hyperparameters $\boldsymbol{\theta}_M$ can be inferred jointly as usual, i.e., it runs as an integrated modeling process. More efficiently, for the deterministic (i.e., noise-free) cases with nested training sets (i.e., $X_T \subseteq X_{T-1} \subseteq \cdots \subseteq X_1$), Kennedy et al. [71] and Le Gratiet et al. [74] have derived that the multi-output modeling process can be decomposed into T successive and independent modeling processes, allowing to infer the hyperparameters for each output separately. The successive process is as follows: (1) build the $(t-1)$-level model $f_{*t-1}(\mathbf{x})$; (2) estimate the hyperparameters $\boldsymbol{\theta}_t$ and the correlation factor a_{t-1} jointly to build $\delta_{*t}(\mathbf{x})$; and (3) obtain the t-level model $f_{*t}(\mathbf{x}) = a_{t-1}f_{*t-1}(\mathbf{x}) + \delta_{*t}(\mathbf{x})$. It is found that this decomposed modeling process is similar to that of the SDM with the only difference occurs in step (2): the SDM estimates a_{t-1} separately, whereas the CoKG treats it as a hyperparameter and estimates it and $\boldsymbol{\theta}_t$ jointly.

For many realistic cases, however, we can only have some observations of the function $f(\mathbf{x})$, which might contain noise; the provided training sets for the outputs might not follow the nested property. Hence, for the general cases considered in this article, we employ two versions of CoKG for the comparison purpose. The first is the integrated CoKG$_a$ that infers all the hyperparameters jointly. The second is the decomposed CoKG$_b$ that infers the hyperparameters for the outputs separately. The CoKG$_b$ is approximately available for general cases here because in the multi-fidelity scenario, the relatively cheap LF outputs can afford many training points; thus the prediction $\hat{f}_{t-1}(\mathbf{x}_{t,i})$ is accurate with a very small variance, i.e., it can be regarded as a deterministic quantity. Similar to the CoKG$_b$, the nlCoKG also runs in a decomposed fashion, see the details in [80].

Another noteworthy thing is that, though inferring the hyperparameters for the outputs separately, the SDM finds the optimal a_{t-1} by solving the auxiliary problem (42), wherein the discrepancy model $\delta_{*t}(\mathbf{x})$ should be refitted frequently. Hence, the SDM modeling process is time-consuming.

In terms of the number of hyperparameters, the ICM-type ($Q = 1$) MTGP model [34] needs to infer $T(T+1)/2$ correlation parameters due to the full rank correlation matrix parameterized by Eq. (22), $d+1$ covariance parameters for the single latent process, and T noise variances for each of the outputs (see Eq. (11)). While for the LMC-type SLFM model [47], this article employs the full rank free-form strategy to parametrize the correlation matrix A_q. Hence, we should infer $QT(T+1)/2$ correlation parameters, $Q(d+1)$ covariance parameters, and T noise parameters. Compared to the SLFM model, the CoMOGP model [42] needs to infer $T(d+1)$ more parameters due to the additional output-specific processes $\{v_t\}_{1\leq t\leq T}$ in Eq. (24). The SST model needs to infer $T(d+1)+T$ hyperparameters in the first stage to build T GP models individually; it then infers $T(d+T+1)+T$ hyperparameters in the second stage to build T GP models individually by the transformed input data. Similarly, the ERC model needs to infer $T(d+1)+T(T-1)/2+T$ hyperparameters to successively build T GP models using one of the s regressor chains. For the four asymmetric MOGPs, the CoKGs [71, 15] and the proposed SDM have the same number of hyperparameters, while the nlCoKG [80] has more hyperparameters due to the complex covariance function (36).

Finally, to reduce the computational complexity of calculating K_M^{-1} for cases with large-scale training data, some sparse approximations that use m ($m \ll N$) inducing points to approximate the large covariance matrix, or use the variational inference to approximate the lower bound of $p(\mathbf{y})$, have been applied to MOGPs [90, 60, 56, 42, 61], reducing the computational complexity to $\mathcal{O}(m^2N)$. Since this article tests the MOGPs with moderate training sizes ($\approx 10^3$), we thus do not implement these sparse approximations.

6. Symmetric modeling experiments

In the symmetric scenario where the T outputs are of equal importance and have the same training size $n_1 = \cdots = n_T = n$, we intend to assess the capability

of various symmetric MOGPs to improve the predictions of all the outputs. The SOGP is also involved in the comparisons to act as a baseline.

We first apply these symmetric MOGPs to three symmetric examples including two analytical examples and a realistic dataset from 1D to 6D with $T = 2$, since the two-output scenario is popular in research works. In this part, we first investigate which type of training data (heterotopic vs. isotopic) is more beneficial for the symmetric MOGPs, followed by the analysis of the characteristics and differences of the symmetric MOGPs. We then study the impact of training size on the performance of symmetric MOGPs. Finally, the symmetric MOGPs are applied to a complicated 21D realistic dataset with $T = 4$.

6.1. Alternative MOGPs and performance measurement

The six alternative symmetric MOGPs include MTGP, SLFM, CoMOGP, CONV, SST and ERC in Table 1 [5]. In the experiments below, we perform data pre-processing by scaling the input domain to $[0, 1]^d$ and normalizing each of the outputs to zero mean and unit variance. These MOGPs follow the structures described in section 5. Particularly, for the LMC-type SLFM model in (16), we set $Q = T$, and compare it to the ICM-type MTGP to see the impact of Q. For the CoMOGP model in (24), we set $Q = 1$ to see the impact of output-specific terms $\{v_t\}_{1\leq t\leq T}$ when compared to MTGP. For the ERC model, we have $s = T!$ regressor chains.

Regarding model parameter settings, the length scales $\{l_i\}_{1\leq i\leq d}$ and the signal variance σ_f^2 in k_{SE} are initialized to 0.5, the noise variances $\{\sigma_{s,t}^2\}_{1\leq t\leq T}$ are initialized to 0.01, and the lower triangular matrix L in Eq. (22) is initialized with the diagonal elements as one and the remaining elements as zero, i.e., A_q is initialized as an identity matrix. Regarding hyperparameters learning, we use the *minimize* function that adopts a conjugate gradient optimization algorithm in the GPML toolbox [6], with a maximum number of iterations as 500. Though these MOGPs have different model structures, i.e., different multi-output covariance functions as well as different number of hyperparameters, we assign the same initial values to the same hyperparameters and infer them with the same optimization settings for making a fair comparison.

Finally, to assess the model accuracy, we adopt the relative average absolute error (RAAE) criterion (also known as standardized mean squared error, SMSE [4])

$$\text{RAAE} = \frac{\sum_{i=1}^{n_{test}} \mid f(\mathbf{x}_i) - \hat{f}(\mathbf{x}_i) \mid}{n_{test} \times \text{STD}}, \tag{44}$$

where n_{test} denotes the number of test points, and STD stands for the standard deviation of function values at the test points. The closer the RAAE value approaches zero, the more accurate the model is.

[5] The asymmetric MOGPs are not included here, because most of them employ a decomposed modeling process, which is unsuitable for the symmetric case.

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

6.2. Typical symmetric modeling with $T = 2$

In order to assess the performance of different symmetric MOGPs, we adopt three examples with different characteristics. The first 1D Toy example has two outputs respectively expressed as

$$f_1(x) = 1.5(x + 2.5)\sqrt{(6x - 2)^2 \sin(12x - 4) + 10}, \qquad x \in [0, 1], \tag{45a}$$

$$f_2(x) = (6x - 2)^2 \sin(12x - 4) + 10, \qquad x \in [0, 1]. \tag{45b}$$

It is found that the output f_1 is a nonlinear transformation of the output f_2. As shown in Fig. 7(a), the two outputs in this example are highly correlated with the Pearson correlation coefficient of $r = 0.95$.

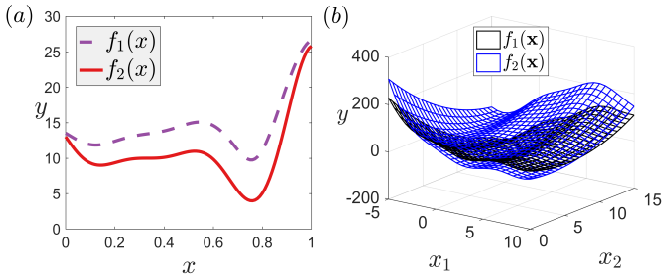

Figure 7: The two outputs of (a) the Toy example and (b) the Branin example.

We then employ a 2D Branin example with two outputs defined in $\Omega_2 \in [-5, 10] \times [0, 15]$ respectively expressed as

$$f_1(\mathbf{x}) = (x_2 - \frac{3}{4\pi}x_1^2 + \frac{4}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 2x_1 - 9x_2 + 32, \tag{46a}$$

$$f_2(\mathbf{x}) = (x_2 - \frac{5.1}{4\pi}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10. \tag{46b}$$

As shown in Fig. 7(b), the output f_1 is a slightly modified Branin function plus a linear discrepancy term, and the output f_2 is the original non-stationary Branin function. Compared to the 1D example, the two outputs in the Branin example are lowly correlated with the Pearson coefficient of $r = 0.67$.

We finally apply the six symmetric MOGPs to a practical aircraft design example. This example selects six important input variables and calculates two aeroengine outputs (gross thrust and net thrust). We use the Matlab routine *lhsdesign* to generate 1000 points and evaluate the two outputs at each point by the design software. Using the 1000 points, it is detected that the correlation coefficient between the gross thrust and the net thrust is $r = 0.87$.

Table 2 shows the sampling configurations for symmetric MOGPs on the three examples. In this table, n denotes the number of training points for each output, since for the symmetric examples we assume $n_1 = n_2 = n$; n_{test} denotes the number of test points used to calculate the RAAE value of the

Table 2: Sampling configurations for symmetric MOGPs on three symmetric examples.

Examples	d	T	$n = n_1 = n_2$				n_{test}	heterotopic vs. isotopic
Toy	1	2	4	6	8	10	100	$n = 6$
Branin	2	2	10	20	40	60	5000	$n = 20$
Aircraft	6	2	20	60	100	140	500	$n = 60$

model for each output. In what follows, we first investigate the performance of symmetric MOGPs using heterotopic training points and isotopic training points, respectively, on the three examples. For the heterotopic training case, we use the function *lhsdesign* to generate the training points for the outputs separately. For the isotopic case, both the outputs f_1 and f_2 use the training sets generated in the heterotopic case for f_2. The training size for the heterotopic vs. isotopic comparison is set as $10d$, because Loeppky et al. [91] has provided reasons and evidence supporting that this initial training size is able to build an effective initial GP model. But since ten points for the 1D Toy example are too many, we set the size as $6d$. Thereafter, the impact of training size n on the performance of symmetric MOGPs is studied by testing the models with another three training sizes that are less or larger than $10d/6d$.

Besides, since the performance of these MOGPs vary for different training sets [29], we statistically test them with 100 different training sets for each sampling configuration. For the analytical Toy and Branin examples, the 100 training sets are generated by the function *lhsdesign*. For the Aircraft dataset, the points in each of the 100 sets are randomly selected from the original dataset. The randomness and the 100 repetitions enlarge the diversity of training sets for thoroughly showcasing the performance of different symmetric MOGPs.

6.2.1. Heterotopic vs. isotopic

Fig. 8 depicts the boxplots of the RAAE values of six symmetric MOGPs and the SOGP using heterotopic/isotopic training points on the Toy example with $n = 6$, the Branin example with $n = 20$ and the Aircraft example with $n = 60$, respectively. These boxplots in compact formating describe how the RAAE values of the models vary over 100 training sets. The bottom and top of each box are the lower and upper quartile values of the RAAEs, the dot circles represent the median RAAE values, the open symbols (square and triangle) represent the average RAAE values, the vertical lines (whiskers) extended from the end of the box represent 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 $+$ and $\times$ symbols represent the outliers that beyond the limit of the vertical lines.

The results in Fig. 8 indicate that the simple transformation models SST and ERC, which treat outputs as inputs, seem to be insensitive to the type of training data. This is induced by the decomposed modeling processes that do not consider the fusion of training data from all the outputs. On the contrary,

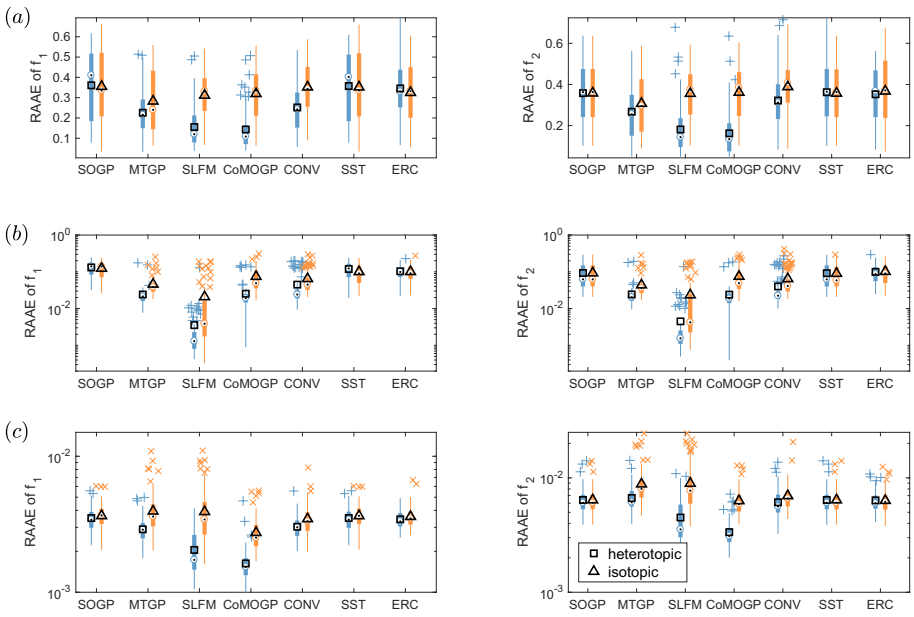

Figure 8: Comparison of different symmetric MOGPs using heterotopic/isotopic training points on (a) the Toy example with $n = 6$, (b) the Branin example with $n = 20$ and (c) the Aircraft example with $n = 60$, respectively.

we observe that *the four Bayesian MOGPs perform better using heterotopic training points*. The core idea behind the multi-output modeling is to transfer information as much as possible across outputs. To maximize the *information diversity*, the outputs are suggested to have unique information, e.g., different training sets. As a result, the MOGP can transfer much information across the outputs when learning them simultaneously. The isotopic case, however, has the same training set for all the outputs, which reduces the diversity of available information. Particularly, given the noise-free observations ($\sigma^2_{s,t} = 0$) at the isotopic training points for all the outputs, Bonilla et al. [34] has pointed out that the predictions of MTGP at a test point $\mathbf{x}_*$ in (14a) are degenerated to

$$\hat{\mathbf{f}}(\mathbf{x}_*) = \begin{bmatrix} \mathbf{k}_*^\intercal K(\overline{X}, \overline{X})^{-1}\mathbf{y}_1 \\ \vdots \\ \mathbf{k}_*^\intercal K(\overline{X}, \overline{X})^{-1}\mathbf{y}_T \end{bmatrix}. \tag{47}$$

It is found that in this case, the prediction for output f_t only depends on the observations $\mathbf{y}_t$, leading to the so-called *transfer cancellation* or *autokrigeability* [92]. From another point of view, the isotopic case can lead to the symmetric Markov property of covariance functions [93] as

$$\text{cov}[y_t(\mathbf{x}'), y_{t'}(\mathbf{x})|y_t(\mathbf{x})] = 0, \forall \mathbf{x} \neq \mathbf{x}'. \tag{48}$$

This property means that if we have already known $y_t(\mathbf{x})$, then observing $y_{t'}(\mathbf{x})$

gives no information to help predict $y_t(\mathbf{x}')$. Note that even in the isotopic case with noise-free observations, the same covariance hyperparameters shared in (47) can still achieve some kind of information sharing across outputs. That is why some of the four Bayesian MOGPs, e.g., the MTGP, outperform the SOGP on the Toy and Branin examples using isotopic training data. On the contrary, the transfer cancellation will not hold in the heterotopic case. Consequently, most of the four Bayesian MOGPs using heterotopic training data perform much better than the SOGP on all the three examples due to the augmented information diversity.

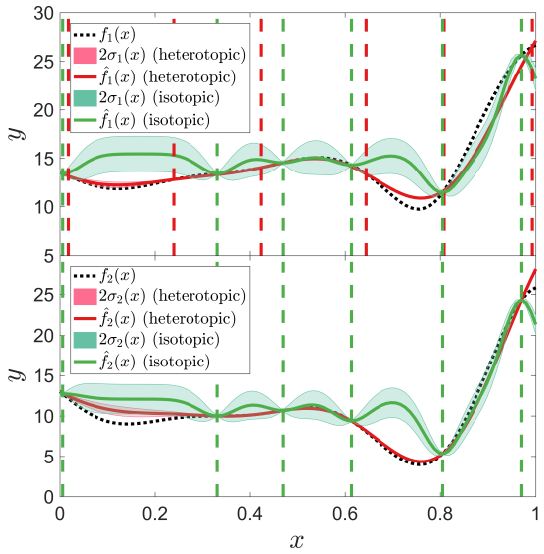

Figure 9: Illustration of heterotopic vs. isotopic on the Toy example using the MTGP model.

Fig. 9 illustrates a run of heterotopic vs. isotopic on the Toy example using the MTGP model, where the dot lines represent the locations of training points. Particularly, the green dot lines represent the isotopic training sets, and they are same for the two outputs. On the other hand, the red dot lines for f_1 and the green dot lines for f_2 represent the heterotopic training sets. We can see that the isotopic training sets deliver poor information about the two outputs because of the same locations. Consequently, the MTGP infers poor hyperparameters with small length scales, leading to no improvement of predictions in comparison to the SOGP. However, the heterotopic case is different. We can imagine that the augmented information diversity brought by the heterotopic training sets is in some way equivalent to increasing the training size for each output. As a result, though each output only has six points, the MTGP infers good hyperparameters and improves the SOGP predictions significantly by fusing the heterotopic training points together.

In conclusion, if the physical problem allows to simulate the outputs separately, it is recommended to generate heterotopic training data to improve

the information diversity for better hyperparameter learning and finally better multi-output modeling.

6.2.2. Comparison of symmetric MOGPs

In this section, we intend to investigate the impact of various MOGP structures that employ different covariance functions (ICM type, LMC type or extended LMC type) and different modeling processes (integrated or decomposed), on the modeling performance. To make a fair comparison, all the symmetric MOGPs follow the parameter settings in section 6.1 for learning hyperparameters and use the heterotopic training data.

We focus on the analysis of the RAAE results of different models using the heterotopic training points in Fig. 8. To this end, we employ the Nemenyi post-hoc test to check the statistical significance of the differences between the models. Particularly, the graphical presentation introduced in [94] is used to report the statistical results. In this diagram, the models are placed along a horizontal axis according to their average ranks. Then, a critical difference (CD) is calculated as the minimal difference in average ranks to decide whether two models are significantly different or not. Finally, the models are classified into several groups, each of which connects together the models that are not significantly different. Fig. 10 depicts the comparison of different models for the two outputs of the three symmetric examples using the Nemenyi test. Note that we use a 0.05 confidence level to calculate the CD in this article.

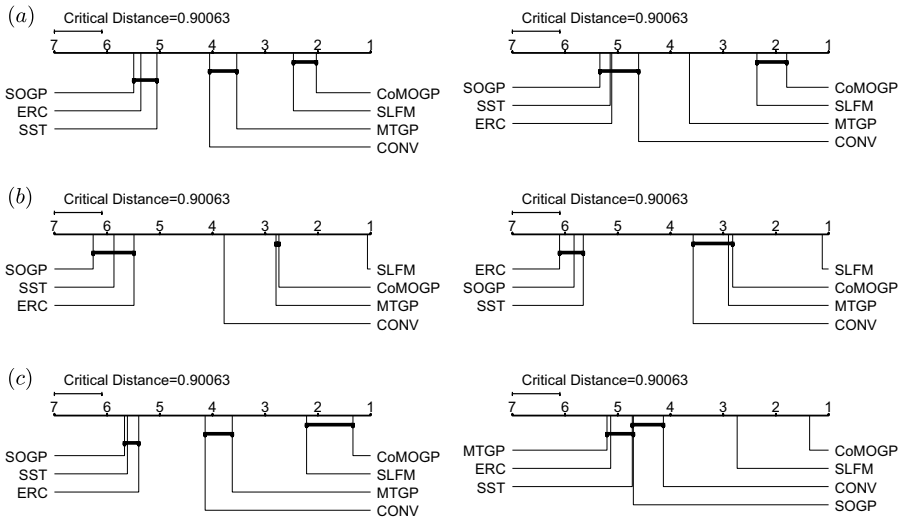

Figure 10: Comparison of different symmetric MOGPs for the two outputs of (a) the Toy example, (b) the Branin example and (c) the Aircraft example using the Nemenyi test. The first column lists the test results for f_1 and the second column for f_2.

The first observation from the statistical results in Fig. 10 is that *the simple transformation models SST and ERC can not significantly improve over the*

SOGP on the three symmetric examples. The poor results of SST and ERC are attributed to the use of the GP model, a very strong base regressor. It has been pointed out that the SST and ERC are able to decrease the model bias at the cost of increasing the model variance by treating the outputs as inputs [64]. If we choose some weak base regressors, e.g., the ridge regression [95], the two models are found to be capable of significantly improving over the single-output models, see the comparative results in [64]. However, for the GP model studied in this article, it is a very strong model wherein the NLML expression (8) itself automatically achieves a bias-variance trade-off [4]. Therefore, due to the capability of well exploiting the observed information, the GP is hard to be improved in multi-output scenarios by using the simple SST and ERC strategies. Besides, the decomposed modeling process limits the performance of SST and ERC, since it does not fuse the information from all the training points of the outputs.

The second observation is that *the four Bayesian symmetric MOGPs (MTGP, SLFM, CoMOGP and CONV) usually significantly outperform the SOGP on the three symmetric examples.* Compared to the SST and ERC that have no clear mechanism to interpret the output correlations, the four Bayesian MOGPs explicitly take into account the output correlations in the correlation matrices $\{A_q\}_{1\leq i\leq Q}$ for effective multi-output modeling.

The final observation is that *the increase of Q ($SLFM$) or the consideration of additional output-specific terms ($CoMOGP$) is able to improve over the MTGP model.* Increasing of Q in SLFM offers more latent functions with different characteristics to enhance the expressive power of the model and transfer more shared information across the outputs. Similarly, the additional output-specific terms in CoMOGP explicitly decomposes the outputs into common and specific features, which make the modeling process easier [7]. As for the CONV model, though containing individual parameters for each output in Eq. (26), it yields a comparable performance to the MTGP model.

6.2.3. Impact of training size

According to the sampling configurations in Table 2, Fig. 11 investigates the impact of training size n on the performance of four MOGPs and the SOGP on three examples. Note that we use the heterotopic training points, and the SST and the ERC are not involved here due to their poor results in Fig. 10 .

It is first observed that all the symmetric MOGPs and the SOGP yield better predictions with the increase of training size (i.e., more information about the outputs). These results are expectable since it has been proved in chapter 7 of [4] that the hyperparameters will be inferred more accurately and the GP predictions will converge to the underlying function values with the increase of training size.

Additionally, we can see that the four symmetric MOGPs often outperform

[7] Note that, similar to the SLFM, we can further improve the performance of the CoMOGP by using $Q > 1$ common processes at the cost of inferring more hyperparameters.

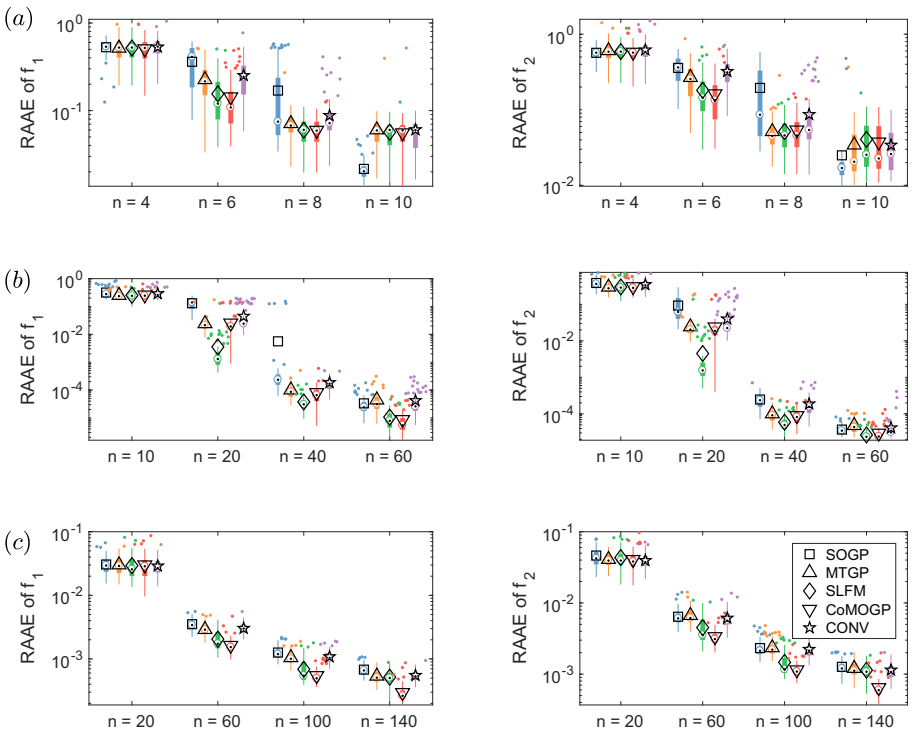

Figure 11: Impact of the training size n on the performance of different symmetric MOGPs on (a) the Toy example, (b) the Branin example and (c) the Aircraft example, respectively.

the SOGP with different training sizes. But they are found to perform similarly to or worse than the SOGP with extreme training sizes. For instance, the MOGPs perform worse than the SOGP for the two outputs of the Toy example with $n = 4$ and for the first output of the Aircraft example with $n = 10$. Likewise, the MTGP and the CONV perform worse than the SOGP on the Branin example with $n = 60$; all the MOGPs except the CoMOGP have deteriorated to be close to the SOGP on the Aircraft example with $n = 140$; and more seriously, all the MOGPs perform worse than the SOGP on the Toy example with $n = 10$.

The poor performance of the symmetric MOGPs using a few training points is induced by the *information sparsity*. If the training size n is too small, the model can not capture the primary features of the outputs. As a result, modeling the outputs jointly can gain little benefits, sometimes it on the contrary may lead to poor predictions. A too large training size n, however, will also harm the performance of symmetric MOGPs due to the *discrepancy* between the shared information and the exact information. For output f_t, the information transferred from other outputs, however, is more or less different from the exact information itself. In cases with limited information about f_t, the additional information transferred from related outputs surely enhance our knowledge about

f_t. But in cases with abundant information about f_t, the transferred information may lead to poor predictions because of the discrepancy between this kind of information and the exact one.

Finally, from the foregoing discussions, we empirically conclude that in the symmetric scenarios, *the four Bayesian symmetric MOGPs outperform the SOGP with moderate training sizes* (*e.g., around* $10d$ $(d > 1)$ *or* $6d$ $(d = 1)$), *but perform worse with extreme training sizes.*

6.3. Robot inverse dynamic example with $T = 4$

Apart from the above three $T = 2$ examples, we finally apply the symmetric MOGPs to a more complicated and high-dimensional realistic example. This example is an inverse dynamic model of a 7-degree-of-freedom anthropomorphic robot arm [96]. It has 21 input variables including 7 joints positions, 7 joint velocities and 7 joint accelerations; the corresponding outputs are 7 joint torques. We attempt to approximate the 2nd, 3rd, 4th and 7th joint torques using the symmetric MOGPs. The Robot dataset contains 48933 data points [42] and it is found that the 2nd torque is negatively correlated with the other three torques, the 2nd and 4th torques have the lowest correlation of $r = -0.57$, whilst the 4th and 7th torques have the highest correlation of $r = 0.96$.

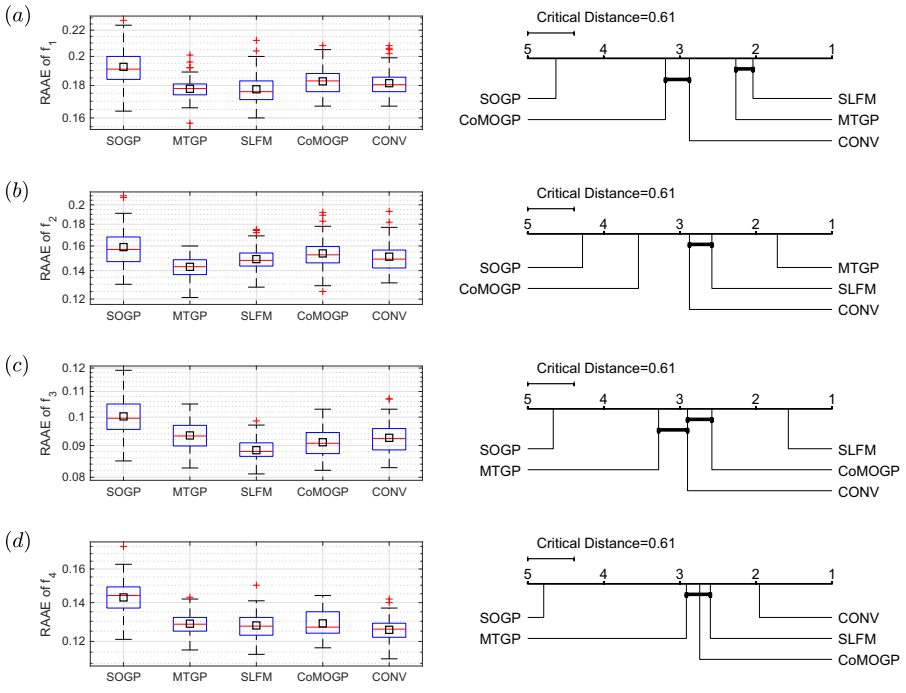

Figure 12: The RAAE values (left column) and the Nemenyi test results (right column) of symmetric MOGPs for the four outputs of the Robot example.

For the complicated Robot example, we follow the $10d$ rule to choose 210 heterotopic training points from the original dataset for each of the four outputs. Similarly, we have 100 different training sets, each of which is randomly selected from the original dataset. Besides, we leave a separate test set containing 4449 data points to calculate the RAAE.

Fig. 12 depicts the RAAE values as well as the Nemenyi test results of the four symmetric MOGPs for the four outputs of the Robot example. We can see that all the four MOGPs outperform the SOGP significantly for the four outputs. Among them, the SLFM produces the best performance, since it is always one of the two best MOGPs for each of the four outputs. Besides, we clearly see that the MOGPs are much more time-consuming than the SOGP, since they fuse all the training points in a single covariance matrix and have many hyperparameters. For instance, a single run of the SLFM takes about 780 s, while a run of the SOGP for the four outputs only needs 39 s.

7. Asymmetric modeling experiments

In the asymmetric multi-fidelity scenario, the intent is to use the related $T-1$ LF outputs to enhance the modeling of the HF output with the training sizes as $n_1 > \cdots > n_T$. This section first studies four asymmetric MOGPs, including CoKG_a, CoKG_b, nlCoKG and SDM in Table 1, on three asymmetric examples with two-level fidelity. The two-level fidelity scenario is by far the most popular of the research works in the literature. Besides, the four Bayesian symmetric MOGPs are also employed for the comparison purpose. We first compare the symmetric/asymmetric MOGPs to see their pros and cons in multi-fidelity scenarios, followed by the study of the impact of HF training size. Finally, these MOGPs are applied to an example with three-level fidelity to assess their performance in the general hierarchical multi-fidelity scenario.

For the asymmetric MOGPs, the correlation factor a_{t-1} is initialized to 1. For the nlCoKG model in (35), we calculate the posterior mean and variance at $\mathbf{x}_*$ by Monte Carlo integration of Eq. (37) using 1000 points sampled from $p(f_{*t-1}(\mathbf{x}_*))$. Besides, in the SDM modeling process, we solve the auxiliary optimization problem (42) with a in the range $[-5, 5]$. Other parameters keep the same as that in the above symmetric experiments.

7.1. Typical asymmetric modeling with two-level fidelity

This section investigates the performance of four asymmetric MOGPs and four symmetric MOGPs on three multi-fidelity examples, each of which has an expensive HF output and a cheap LF output. The first two examples are the Toy example and the Branin example used before. Here, we regard f_1 as the inexpensive LF output and f_2 as the expensive HF output. The third realistic Airfoil example comes from [79] that calculates the lift coefficient of an airfoil under different flight conditions and geometry parameters. The Airfoil data is generated based on six most important design variables selected from 52 design variables including the geometry parameters, the speed and the angle of attack

Table 3: Sampling configurations for symmetric/asymmetric MOGPs on three asymmetric examples.

Examples	d	T	$n_h = n_2$				$n_l = n_1$	n_{test}^h	n_{test}^l
Toy	1	2	4	5	6	8	12	100	100
Branin	2	2	5	10	20	40	60	5000	5000
Airfoil	6	2	20	40	60	140	200	200	1500

[97]. The lift coefficient is calculated using two solvers with different levels of fidelity, resulting in 365 HF points and 1996 LF points. Finally, it is detected from the dataset that the Pearson correlation coefficient between the HF and LF solvers is $r = 0.90$.

Table 3 shows the sampling configurations for symmetric/asymmetric MOGPs on the three asymmetric examples. The symbol n_h denotes the number of HF training points, n_l denotes the number of LF training points, and n_{test}^h and n_{test}^l denote the numbers of HF and LF test points, respectively. The large amount of LF points indicate that the LF output can be well approximated. Similar to the above symmetric experiments, the training set for the Toy and Branin examples is generated by the *lhsdesign* function and has 100 instances. Note that the 100 instances of each sampling configuration for the Airfoil example are randomly selected from the original dataset.

7.1.1. Comparison of different MOGPs

Fig. 13 depicts the RAAE values of different symmetric/asymmetric MOGPs and the SOGP on the Toy example with $n_h = 6$, the Branin example with $n_h = 20$, and the Airfoil example with $n_h = 60$, respectively. Besides, the corresponding right sub-figures provide the Nemenyi test results of these MOGPs.

We first observe that *most of the symmetric/asymmetric MOGPs significantly outperform the SOGP on the three asymmetric examples*. For example, all the MOGPs significantly outperform the SOGP on the Toy example; all of them except the SDM significantly outperform the SOGP on the Airfoil example; six out of the eight MOGPs perform significantly better than the SOGP on the Branin example.

Besides, we observe that *with the decrease of output correlations, the asymmetric MOGPs using decomposed modeling process outperform the symmetric MOGPs*. The integrated modeling processes of the symmetric MOGPs enable them to mimic the output behaviors from one another by bidirectional information transfer between the outputs. Different from the symmetric scenario where the two outputs have similar amount of training information (e.g., $n_1 = n_2$), the asymmetric scenario here, however, has an information asymmetry property ($n_h(n_2) < n_l(n_1)$). As a result, the LF output contributes to most of the information in the correlation matrix K_M. Hence, through transferring information bidirectionally between outputs, the symmetric MOGPs mainly mimic the LF behaviors to represent the HF output. If the HF/LF outputs are highly correlated, the symmetric MOGPs are expected to have good predictions. For

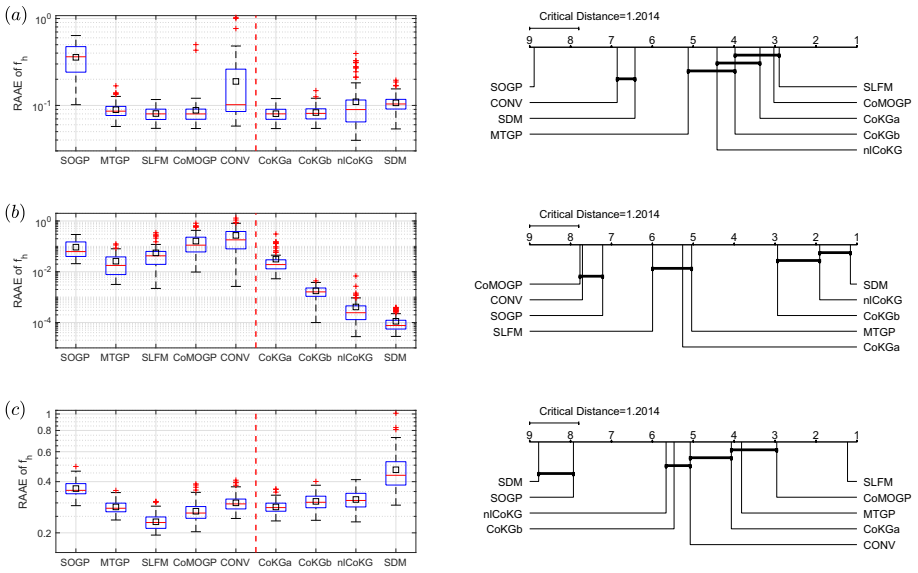

Figure 13: The RAAE values (left column) and the Nemenyi test results (right column) of symmetric and asymmetric MOGPs on (a) the Toy example with $n_h = 6$, (b) the Branin example with $n_h = 20$ and (c) the Airfoil example with $n_h = 60$, respectively.

instance, two of the three best MOGPs on the Toy example ($r = 0.95$) are symmetric MOGPs, see, e.g., the CoMOGP run in Fig. 14(a); and all the three best MOGPs on the Airfoil example ($r = 0.90$) are symmetric MOGPs. However, if the HF/LF outputs have a low correlation, the HF predictions mainly learnt from the LF output may be far away from the exact value. This issue becomes more serious for those symmetric MOGPs with a strong expressive power, i.e., being capable of learning more from the LF output. For instance, the CoMOGP has the poorest predictions on the Branin example ($r = 0.67$), see an illustration run in Fig. 14(b). Finally, it is found that different from other asymmetric MOGPs, the CoKG$_a$ performs similarly to the symmetric MOGPs on the three examples due to the integrated modeling process.

On the contrary, the remaining three asymmetric MOGPs (CoKG$_b$, nlCoKG and SDM) unidirectionally and purely transfer all the LF information to the HF output due to the decomposed modeling process. To avoid purely using the LF behaviors to represent the HF output, they additionally consider a discrepancy term $\delta_t(\mathbf{x})$ as well as a correlation factor a_{t-1} to adjust the HF predictions. As a result, the three asymmetric MOGPs significantly outperform the other models on the complex Branin example, see, for example, the CoKG$_b$ run in Fig. 14(b). A question arises that the CoKG$_a$ also contains a discrepancy term but cannot yield such good results on the Branin example. We think it is attributed to the integrated modeling process where the common and discrepancy information is fused in K_M. If the shared information is not accurate for f_h, which occurs for symmetric MOGPs on this example, the quality of discrepancy information will

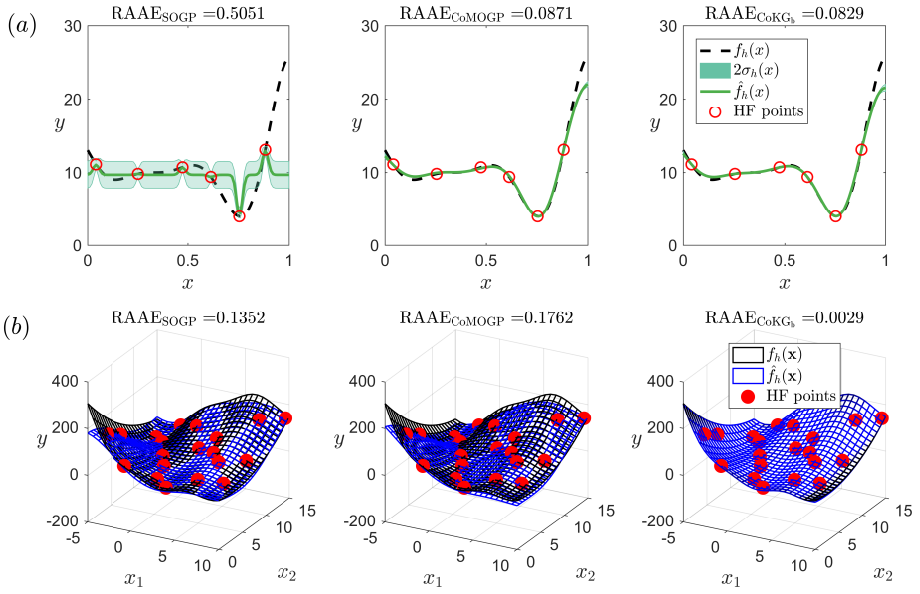

Figure 14: Illustration of the SOGP, the CoMOGP and the CoKG$_b$ on (a) the Toy example with $n_h = 6$ and (b) the Branin example with $n_h = 20$.

be affected. Hence, the decomposed modeling process here not only reduces the computational complexity, but also provides benefits for the prediction quality.

Besides, for the other two examples with highly correlated outputs, there is no significant difference between the symmetric MOGPs (SLFM and CoMOGP) and the asymmetric MOGPs (CoKG$_a$ and CoKG$_b$) on the 1D Toy example; but the asymmetric MOGPs perform significantly worse than the SLFM on the 6D Airfoil example. Among the three asymmetric MOGPs, the CoKG$_b$ and the nlCoKG significantly outperform the simple SDM on two examples. As for the complex Branin example, the nlCoKG outperforms the CoKG$_b$ since it has the capability to capture the nonlinear correlations over the whole domain. Interestingly, the simple SDM offers a comparable performance to the nlCoKG on the Branin example.

Finally, note that the decomposed modeling processes in CoKG$_b$, nlCoKG and SDM gain benefits from the information asymmetry property, i.e., the inexpensive LF output owns many more training points than the expensive HF output. If the information asymmetry is no longer hold, for example, it degenerates to the symmetric case with $n_l = n_h$, the decomposed modeling process will yield poor results because of the poor quality of LF predictions. Taking the Branin case with $n_l = n_h = 20$ for example, we have $\text{RAAE}_{\text{nlCoKG}} = 0.0923$ and $\text{RAAE}_{\text{SLFM}} = 0.0045$. That is why we did not include the asymmetric MOGPs in the symmetric modeling experiments in section 6.

7.1.2. Impact of HF training size

Fig. 15 investigates the impact of HF training size n_h on the RAAE values of different MOGPs on the three asymmetric examples according to the sampling configurations in Table 3.

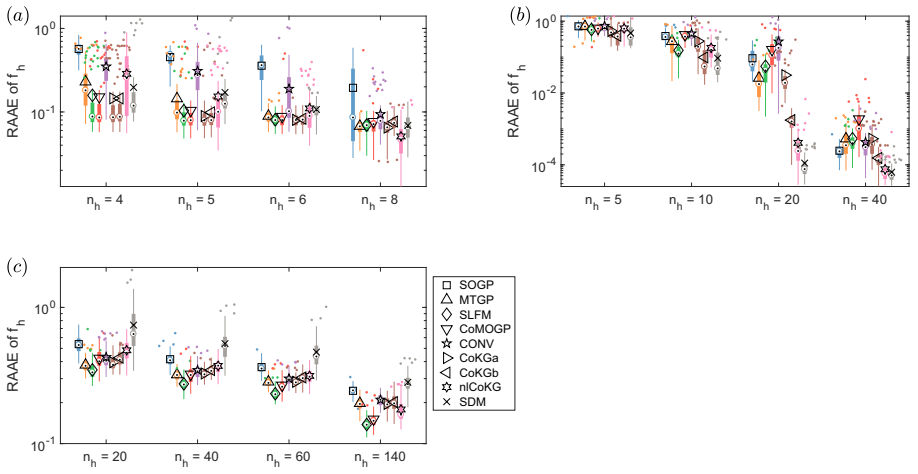

Figure 15: The impact of HF training size n_h on the RAAE values of different MOGPs on (a) the Toy example, (b) the Branin example and (c) the Airfoil example, respectively.

Different from the symmetric results in Fig. 11, we here find that the MOGPs can generally improve the SOGP predictions with a small HF training size, e.g., $n_h = 4$ for Toy, $n_h = 5$ for Branin and $n_h = 20$ for Airfoil. This is because the LF output has already provided sufficient related information to help learn the HF output in the asymmetric scenario. Besides, with the increase of n_h, the MOGPs, especially the symmetric MOGPs, begin to lose their ability to improve over the SOGP. For example, the median RAAE values of most MOGPs are close to that of the SOGP with $n_h = 8$ on the Toy example. More seriously, the symmetric MOGPs together with the CoKG$_a$ perform worse than the SOGP with $n_h = 40$ on the Branin example. The reason is the same for the deterioration of symmetric MOGPs with large training sizes in Fig. 11.

Additionally, it is observed that the nlCoKG using a space-dependent correlation term z_{t-1} in Eq. (35) usually performs worse than the CoKG$_b$ using a constant correlation term a_{t-1} in Eq. (31) on the three asymmetric examples with small HF training sizes. The reason is that the nlCoKG is hard to accurately capture the nonlinear HF/LF correlations with a few HF training points. But the nlCoKG begins to outperform the CoKG$_b$ with the increase of n_h, especially on the Branin example with complex output correlations.

Finally, from the foregoing discussions, we empirically conclude that in the multi-fidelity asymmetric scenarios, *the symmetric/asymmetric MOGPs outperform the SOGP with small and moderate HF training sizes* (*e.g., conservatively, less than about* $10d$ $(d > 1)$ *or* $6d$ $(d = 1)$)*, but perform worse with large HF training sizes.*

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

As stated before, the multi-fidelity scenario has a hierachical structure with $n_1 > \cdots > n_T$. Hence, we finally applied the MOGPs to a realistic example with three-level fidelity. This example is a 2D stochastic incompressible flow (SIFlow) passing a circular cylinder under a random inflow boundary condition [98], with the goal of modeling the 0.6-superquantile risk of the base pressure coefficient $\mathcal{R}_{0.6}(C_{BP})$ with the condition of Reynolds number $Re = 100$. The output should be simulated in both the physical space wherein the incompressible flow field is governed by the Navier-Stokes equations, and the probability space wherein three simulators of different precisions are used. The computing time of the HF simulator is about 3 times the medium fidelity (MF) simulator, and is about 15 times the LF simulator. The SIFlow dataset comprises 30 HF data points, 99 MF data points and 357 LF data points, which are simulated in parallel on one rack of IBM BG/Q (16384 cores) [98].

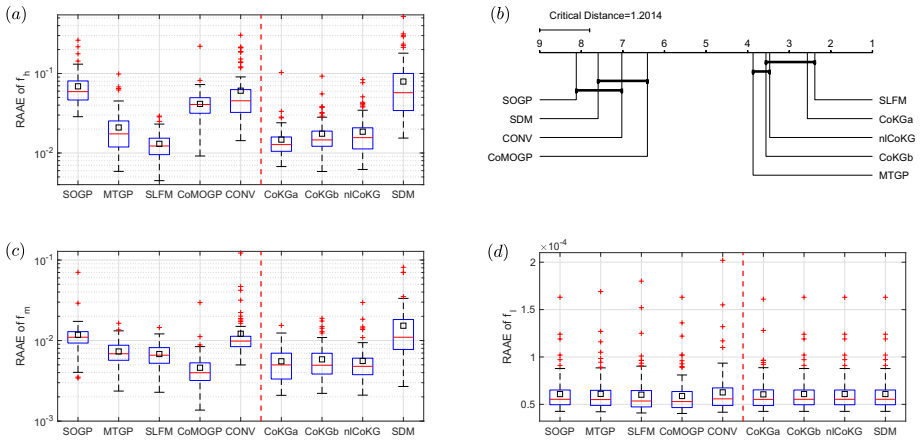

Figure 16: The RAAE values of symmetric and asymmetric MOGPs for (a) the HF output, (c) the MF output and (c) the LF output of the SIFlow example, respectively. Besides, the Nemenyi test results of the MOGPs on the HF output are provided in (b).

For the SIFlow example with three-level fidelity, we test different MOGPs with 10 HF points, 30 MF points and 100 LF points. Each of the training set has 100 instances randomly selected from the original dataset. Fig. 16 depicts the RAAE values and the Nemenyi test results of different MOGPs on the SIFlow example. Note that though our intent is to improve the prediction quality of the HF output, we still provide the results of MOGPs for the MF and LF outputs in this figure to show the gradual improving process from the LF output to the HF output.

In such a hierarchical scenario, the symmetric MOGPs are capable of improving the HF output, since they fuse the information of all the outputs and transfer knowledge between each other. But the improvements brought by the symmetric MOGPs seem to be weaker than most of the asymmetric MOGPs,

which are specifically designed for the hierarchical scenario. For instance, except the CoMOGP, the other three symmetric MOGPs yield larger RAAE values than three asymmetric MOGPs for the MF output. As for the final HF output, though the SLFM shows a competitive performance, the other symmetric MOGPs perform significantly worse than most of the asymmetric MOGPs. Finally, in the four asymmetric MOGPs, the simple SDM can not improve over the SOGP for both the MF and HF outputs.

8. Conclusions

This article attempts to provide insights into the state-of-the-art MOGPs, which are promising for multi-output approximation in terms of improving prediction quality. We classify them into two main categories as (1) symmetric MOGPs and (2) asymmetric MOGPs. We thoroughly examined the performance of ten representative MOGPs on eight examples in the symmetric and asymmetric scenarios. Eight out of the ten MOGPs are found to be promising in terms of improving the SOGP predictions for multiple outputs. Though according to the no-free lunch theorem [99], no one in the eight MOGPs always outperforms the others. From the review and the numerical experiments, we can provide some recommendations for the use of these MOGPs and highlight potential directions for further research.

In the symmetric scenario where the intent is to improve the predictions of all the outputs jointly, we considered $n_1 = \cdots = n_T$ and investigated six representative symmetric MOGPs on four examples using different types of training points, different training sizes and different output sizes. Based on the numerical experiments, we have the following findings:

- If the physical problem allows to simulate the outputs separately, it is recommended to generate heterotopic training data to improve the information diversity for better multi-output modeling;

- The simple transformation models SST and ERC can not significantly improve over the SOGP. Besides, they are hard to understand and interpret multi-output problems;

- The four Bayesian symmetric MOGPs (MTGP, SLFM, CoMOGP and CONV) usually significantly outperform the SOGP. Among them, the CONV is found to have a performance similar to the ICM-type MTGP. But the increase of Q (SLFM) or the consideration of additional output-specific terms (CoMOGP) is able to improve over the MTGP;

- The four Bayesian symmetric MOGPs outperform the SOGP with moderate training sizes (e.g., around $10d$ ($d > 1$) or $6d$ ($d = 1$)), but perform worse with extreme training sizes.

Moreover, in the multi-fidelity asymmetric scenario where the intent is to extract information from related and inexpensive LF outputs to approximate

the expensive HF output, we considered $n_1 > \cdots > n_T$ and investigated four symmetric MOGPs and four asymmetric MOGPs on four examples, and have the following findings:

- Most of the symmetric/asymmetric MOGPs significantly outperform the SOGP in the multi-fidelity scenario;
- With the decrease of output correlations, the asymmetric MOGPs using decomposed modeling process, e.g., CoKG$_b$ and nlCoKG, outperform the symmetric MOGPs;
- The symmetric/asymmetric MOGPs outperform the SOGP with small and moderate HF training sizes (e.g., conservatively, less than about $10d$ ($d > 1$) or $6d$ ($d = 1$)), but perform worse with large HF training sizes.

Last, based on the qualitative and quantitative analysis of the state-of-the-art MOGPs, we highlight some potential research directions to further improve the performance of MOGPs and extend the available applications. For example, we can devote to (1) effective MOGPs that utilize positive transfer while avoid negative transfer across outputs [48, 62]; (2) efficient MOGPs that handle "Big data" with numerous training points in high dimensions [42, 77]; (3) non-stationary MOGPs that handle, e.g., discontinuous problems [52, 100]; (4) sampling strategies for sequential updating of the MOGPs by considering the output correlations and, particularly, the computing ratios between different fidelity levels in asymmetric scenarios [75, 101]; and (5) MOGP-assisted optimization [102, 103] and uncertainty quantification [104].

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

[1] G. G. Wang, S. Shan, Review of metamodeling techniques in support of engineering design optimization, Journal of Mechanical Design 129 (2007) 370–380.

[2] S. Razavi, B. A. Tolson, D. H. Burn, Review of surrogate modeling in water resources, Water Resources Research 48 (2012) W07401.

[3] H. Liu, S. Xu, X. Wang, Sampling strategies and metamodeling techniques for engineering design: Comparison and application, in: ASME Turbo Expo 2016: Turbomachinery Technical Conference and Exposition, ASME, 2016, pp. V02CT45A019–V02CT45A019.

[4] C. E. Rasmussen, C. K. I. Williams, Gaussian processes for machine learning, MIT Press, 2006.

[5] J. P. Kleijnen, Kriging metamodeling in simulation: A review, European Journal of Operational Research 192 (2009) 707–716.

[6] Y. Wang, B. Chaib-draa, KNN-based kalman filter: An efficient and non-stationary method for gaussian process regression, Knowledge-Based Systems 114 (2016) 148–155.

[7] N. Lawrence, Probabilistic non-linear principal component analysis with gaussian process latent variable models, Journal of Machine Learning Research 6 (2005) 1783–1816.

[8] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, N. de Freitas, Taking the human out of the loop: A review of bayesian optimization, Proceedings of the IEEE 104 (2016) 148–175.

[9] A. O'Hagan, Bayesian analysis of computer code outputs: A tutorial, Reliability Engineering & System Safety 91 (2006) 1290–1300.

[10] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, S. Aigrain, Gaussian processes for time-series modelling, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 371 (2013) 20110550.

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

[12] C. Williams, S. Klanke, S. Vijayakumar, K. M. Chai, Multi-task gaussian process learning of robot inverse dynamics, in: Advances in Neural Information Processing Systems, Curran Associates, Inc., 2009, pp. 265–272.

[13] R. Dürichen, M. A. Pimentel, L. Clifton, A. Schweikard, D. A. Clifton, Multitask gaussian processes for multivariate physiological time-series analysis, IEEE Transactions on Biomedical Engineering 62 (2015) 314–322.

[14] P. M. Zadeh, V. V. Toropov, A. S. Wood, Metamodel-based collaborative optimization framework, Structural and Multidisciplinary Optimization 38 (2009) 103–115.

[15] A. I. Forrester, A. Sóbester, A. J. Keane, Multi-fidelity optimization via surrogate modelling, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 463 (2007) 3251–3269.

[16] T. C. Haas, Multivariate spatial prediction in the presence of non-linear trend and covariance non-stationarity, Environmetrics 7 (1996) 145–165.

[17] J. M. Ver Hoef, R. P. Barry, Constructing and fitting models for cokriging and multivariable spatial prediction, Journal of Statistical Planning and Inference 69 (1998) 275–294.

[18] J.-P. Chiles, P. Delfiner, Geostatistics: Modeling spatial uncertainty, John Wiley & Sons, 1999.

[19] P. Diggle, P. Ribeiro, Model-based geostatistics, Springer, 2007.

[20] J. Baxter, A bayesian/information theoretic model of learning to learn via multiple task sampling, Machine Learning 28 (1997) 7–39.

[21] R. Caruana, Multitask learning, Machine Learning 28 (1997) 41–75.

[22] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering 22 (2010) 1345–1359.

[23] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, G. Zhang, Transfer learning using computational intelligence: A survey, Knowledge-Based Systems 80 (2015) 14–23.

[24] B. Kulis, K. Saenko, T. Darrell, What you saw is not what you get: Domain adaptation using asymmetric kernel transforms, in: Computer Vision and Pattern Recognition, IEEE, 2011, pp. 1785–1792.

[25] M. Kandemir, Asymmetric transfer learning with deep gaussian processes, in: Proceedings of the 32nd International Conference on Machine Learning, PMLR, 2015, pp. 730–738.

[26] P. Wei, R. Sagarna, Y. Ke, Y.-S. Ong, C.-K. Goh, Source-target similarity modelings for multi-source transfer gaussian process regression, in: Proceedings of the 34th International Conference on Machine Learning, PMLR, 2017, pp. 3722–3731.

[27] Z.-H. Han, S. Görtz, R. Zimmermann, Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function, Aerospace Science and Technology 25 (2013) 177–189.

[28] A. Zaytsev, E. Burnaev, Large scale variable fidelity surrogate modeling, Annals of Mathematics and Artificial Intelligence 81 (2017) 167–186.

[29] C. Park, R. T. Haftka, N. H. Kim, Remarks on multi-fidelity surrogates, Structural and Multidisciplinary Optimization 55 (2017) 1029–1050.

[30] H. Borchani, G. Varando, C. Bielza, P. Larrañaga, A survey on multi-output regression, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 5 (2015) 216–233.

[31] R. Ababou, A. C. Bagtzoglou, E. F. Wood, On the condition number of covariance matrices in kriging, estimation, and simulation of random fields, Mathematical Geology 26 (1994) 99–133.

[32] R. M. Neal, Monte carlo implementation of gaussian process models for bayesian regression and classification, arXiv preprint physics/9701026 (1997).

[33] R. B. Gramacy, H. K. Lee, Cases for the nugget in modeling computer experiments, Statistics and Computing 22 (2012) 713–722.

[34] E. V. Bonilla, K. M. A. Chai, C. K. Williams, Multi-task gaussian process prediction, in: Advances in Neural Information Processing Systems, Curran Associates Inc., 2007, pp. 153–160.

[35] B. Rakitsch, C. Lippert, K. Borgwardt, O. Stegle, It is all in the noise: Efficient multi-task gaussian process inference with structured residuals, in: Advances in Neural Information Processing Systems, Curran Associates, Inc., 2013, pp. 1466–1474.

[36] A. G. Journel, C. J. Huijbregts, Mining geostatistics, Academic Press, 1978.

[37] M. Goulard, M. Voltz, Linear coregionalization model: Tools for estimation and choice of cross-variogram matrix, Mathematical Geology 24 (1992) 269–286.

[38] A. M. Schmidt, A. E. Gelfand, A bayesian coregionalization approach for multivariate pollutant data, Journal of Geophysical Research: Atmospheres 108 (2003) STS10.1–STS10.8.

[39] T. R. Fanshawe, P. J. Diggle, Bivariate geostatistical modelling: A review and an application to spatial variation in radon concentrations, Environmental and Ecological Statistics 19 (2012) 139–160.

[40] M. A. Alvarez, L. Rosasco, N. D. Lawrence, et al., Kernels for vector-valued functions: A review, Foundations and Trends® in Machine Learning 4 (2012) 195–266.

[41] P. Goovaerts, Geostatistics for natural resources evaluation, Oxford University Press, 1997.

[42] T. V. Nguyen, E. V. Bonilla, et al., Collaborative multi-output gaussian processes, in: Proceedings of the 13rd Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2014, pp. 643–652.

[43] T. E. Fricker, J. E. Oakley, N. M. Urban, Multivariate gaussian process emulators with nonseparable covariance structures, Technometrics 55 (2013) 47–56.

[44] K. Yu, V. Tresp, A. Schwaighofer, Learning gaussian processes from multiple tasks, in: Proceedings of the 22nd International Conference on Machine Learning, ACM, 2005, pp. 1012–1019.

[45] T. Cohn, L. Specia, Modelling annotator bias with multi-task gaussian processes: An application to machine translation quality estimation, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL, 2013, pp. 32–42.

[46] M. A. Osborne, S. J. Roberts, A. Rogers, S. D. Ramchurn, N. R. Jennings, Towards real-time information processing of sensor network data using computationally efficient multi-output gaussian processes, in: Proceedings of the 7th International Conference on Information Processing in Sensor Networks, IEEE, 2008, pp. 109–120.

[47] Y.-W. Teh, M. Seeger, M. Jordan, Semiparametric latent factor models, in: Workshop on Artificial Intelligence and Statistics 10, 2005, pp. EPFL–CONF–161317.

[48] G. Leen, J. Peltonen, S. Kaski, Focused multi-task learning in a gaussian process framework, Machine Learning 89 (2012) 157–182.

[49] H. Liu, Y.-S. Ong, J. Cai, Y. Wang, Cope with diverse data structures in multi-fidelity modeling: A gaussian process method, Engineering Applications of Artificial Intelligence (2017) 1–34.

[50] A. E. Gelfand, A. M. Schmidt, S. Banerjee, C. Sirmans, Nonstationary multivariate process modeling through spatially varying coregionalization, Test 13 (2004) 263–312.

[51] A. Wilson, Z. Ghahramani, D. A. Knowles, Gaussian process regression networks, in: Proceedings of the 29th International Conference on Machine Learning, Omnipress, 2012, pp. 599–606.

[52] B. Konomi, G. Karagiannis, A. Sarkar, X. Sun, G. Lin, Bayesian treed multivariate gaussian process with adaptive design: Application to a carbon capture unit, Technometrics 56 (2014) 145–158.

[53] B. Konomi, G. Karagiannis, G. Lin, On the bayesian treed multivariate gaussian process with linear model of coregionalization, Journal of Statistical Planning and Inference 157-158 (2015) 1–15.

[54] K. Hayashi, T. Takenouchi, R. Tomioka, H. Kashima, Self-measuring similarity for multi-task gaussian process, in: Proceedings of ICML Workshop on Unsupervised and Transfer Learning, PMLR, 2012, pp. 145–153.

[55] T. Hori, D. Montcho, C. Agbangla, K. Ebana, K. Futakuchi, H. Iwata, Multi-task gaussian process for imputing missing data in multi-trait and multi-environment trials, Theoretical and Applied Genetics 129 (2016) 2101–2115.

[56] M. A. Álvarez, N. D. Lawrence, Computationally efficient convolved multiple output gaussian processes, Journal of Machine Learning Research 12 (2011) 1459–1500.

[57] P. Boyle, M. Frean, Dependent gaussian processes, in: Advances in Neural Information Processing Systems, MIT Press, 2005, pp. 217–224.

[58] A. Melkumyan, F. Ramos, Multi-kernel gaussian processes, in: Twenty-Second International Joint Conference on Artificial Intelligence, AAAI Press, 2011, pp. 1408–1413.

[59] N. D. Lawrence, G. Sanguinetti, M. Rattray, Modelling transcriptional regulation using gaussian processes, in: Advances in Neural Information Processing Systems, MIT Press, 2007, pp. 785–792.

[60] M. Alvarez, N. D. Lawrence, Sparse convolved gaussian processes for multi-output regression, in: Advances in Neural Information Processing Systems, Curran Associates, Inc., 2009, pp. 57–64.

[61] J. Zhao, S. Sun, Variational dependent multi-output gaussian process dynamical systems, Journal of Machine Learning Research 17 (2016) 4134–4169.

[62] N. Wagle, E. W. Frew, Forward adaptive transfer of gaussian process regression, Journal of Aerospace Information Systems 14 (2017) 214–231.

[63] D. Higdon, et al., Space and space-time modeling using process convolutions, in: Quantitative Methods for Current Environmental Issues, Springer, 2002, pp. 37–56.

[64] E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, I. Vlahavas, Multi-target regression via input space expansion: Treating targets as inputs, Machine Learning 104 (2016) 55–98.

[65] D. H. Wolpert, Stacked generalization, Neural Networks 5 (1992) 241–259.

[66] G. Li, S. Azarm, A. Farhang-Mehr, A. Diaz, Approximation of multiresponse deterministic engineering simulations: A dependent metamodeling approach, Structural and Multidisciplinary Optimization 31 (2006) 260–269.

[67] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, Machine Learning 85 (2011) 333–359.

[68] W. Cheng, E. Hüllermeier, K. J. Dembczynski, Bayes optimal multilabel classification via probabilistic classifier chains, in: Proceedings of the 27th International Conference on Machine Learning, Omnipress, 2010, pp. 279–286.

[69] K. Dembczyński, W. Waegeman, W. Cheng, E. Hüllermeier, On label dependence and loss minimization in multi-label classification, Machine Learning 88 (2012) 5–45.

[70] J. Read, J. Hollmén, Multi-label classification using labels as hidden nodes, arXiv preprint arXiv:1503.09022 (2015).

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

[72] D. E. Myers, Matrix formulation of co-kriging, Mathematical Geology 14 (1982) 249–257.

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

[74] L. Le Gratiet, J. Garnier, Recursive co-kriging model for design of computer experiments with multiple levels of fidelity, International Journal for Uncertainty Quantification 4 (2014) 365–386.

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

[76] E. Burnaev, A. Zaytsev, Surrogate modeling of multifidelity data for large samples, Journal of Communications Technology and Electronics 60 (2015) 1348–1355.

[77] P. Perdikaris, D. Venturi, G. E. Karniadakis, Multifidelity information fusion algorithms for high-dimensional systems and massive data sets, SIAM Journal on Scientific Computing 38 (2016) B521–B538.

[78] S. Ulaganathan, I. Couckuyt, F. Ferranti, E. Laermans, T. Dhaene, Performance study of multi-fidelity gradient enhanced kriging, Structural and Multidisciplinary Optimization 51 (2015) 1017–1033.

[79] A. Zaytsev, E. Burnaev, Minimax approach to variable fidelity data interpolation, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, PMLR, 2017, pp. 652–661.

[80] P. Perdikaris, M. Raissi, A. Damianou, N. Lawrence, G. Karniadakis, Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473 (2017) 20160751.

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

[82] K. J. Chang, R. T. Haftka, G. L. Giles, I.-J. Kao, Sensitivity-based scaling for approximating structural response, Journal of Aircraft 30 (1993) 283–288.

[83] N. M. Alexandrov, J. Dennis, R. M. Lewis, V. Torczon, A trust-region framework for managing the use of approximation models in optimization, Structural and Multidisciplinary Optimization 15 (1998) 16–23.

[84] R. Lewis, S. Nash, A multigrid approach to the optimization of systems governed by differential equations, in: 8th Symposium on Multidisciplinary Analysis and Optimization, AIAA, 2000, pp. AIAA–2000–4890.

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

[86] X. Li, W. Gao, L. Gu, C. Gong, Z. Jing, H. Su, A cooperative radial basis function method for variable-fidelity surrogate modeling, Structural and Multidisciplinary Optimization 56 (2017) 1077–1092.

[87] Q. Zhou, Y. Wang, S.-K. Choi, P. Jiang, X. Shao, J. Hu, A sequential multi-fidelity metamodeling approach for data regression, Knowledge-Based Systems 134 (2017) 199–212.

[88] C. Durantin, J. Rouxel, J.-A. Désidéri, A. Glière, Multifidelity surrogate modeling based on radial basis functions, Structural and Multidisciplinary Optimization 56 (2017) 1061–1075.

[89] Z.-H. Zhou, J. Wu, W. Tang, Ensembling neural networks: Many could be better than all, Artificial Intelligence 137 (2002) 239–263.

[90] J. Quinonero-Candela, C. E. Rasmussen, C. K. Williams, Approximation methods for gaussian process regression, in: In Large-Scale Kernel Machines, Neural Information Processing, MIT Press, 2007, pp. 203–224.

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

[92] H. Wackernagel, Multivariate geostatistics: An introduction with applications, Springer Science & Business Media, 2013.

[93] A. O'Hagan, A Markov property for covariance structures, Technical Report Statistics Research Report 98-13, Nottingham University, 1998.

[94] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

[95] A. E. Hoerl, R. W. Kennard, Ridge regression: Biased estimation for nonorthogonal problems, Technometrics 12 (1970) 55–67.

[96] S. Vijayakumar, S. Schaal, Locally weighted projection regression: An o(n) algorithm for incremental real time learning in high dimensional space, in: Proceedings of the 17th International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., 2000, pp. 288–293.

[97] A. Bernstein, E. Burnaev, S. Chernova, F. Zhu, N. Qin, Comparison of three geometric parameterization methods and their effect on aerodynamic optimization, in: Eurogen, 2011, pp. 14–16.

[98] P. Perdikaris, D. Venturi, J. Royset, G. Karniadakis, Multi-fidelity modelling via recursive co-kriging and gaussian–markov random fields, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 471 (2015) 20150018.

[99] D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation 1 (1997) 67–82.

[100] M. Raissi, G. Karniadakis, Deep multi-fidelity gaussian processes, arXiv preprint arXiv:1604.07484 (2016).

[101] H. Liu, Y.-S. Ong, J. Cai, A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design, Structural and Multidisciplinary Optimization (2017) 1–24.

[102] D. Huang, T. Allen, W. Notz, R. Miller, Sequential kriging optimization using multiple-fidelity evaluations, Structural and Multidisciplinary Optimization 32 (2006) 369–382.

[103] K. Swersky, J. Snoek, R. P. Adams, Multi-task bayesian optimization, in: Advances in Neural Information Processing Systems, Curran Associates, Inc., 2013, pp. 2004–2012.

[104] I. Bilionis, N. Zabaras, B. A. Konomi, G. Lin, Multi-output separable gaussian process: Towards an efficient, fully bayesian paradigm for uncertainty quantification, Journal of Computational Physics 241 (2013) 212–239.