Bayesian Treed Multivariate Gaussian Process with Adaptive Design: Application to a Carbon Capture Unit

Bledar Konomi*, Georgios Karagiannis, Avik Sarkar, Xin Sun, and Guang Lin
Pacific Northwest National Laboratory

December 19, 2013

Abstract

Computer experiments are widely used in scientific research to study and predict the behavior of complex systems, which often have responses consisting of a set of non-stationary outputs. The computational cost of simulations at high resolution often is expensive and impractical for parametric studies at different input values. In this paper, we develop a Bayesian treed multivariate Gaussian process (BTMGP) as an extension of the Bayesian treed Gaussian process (BTGP) to model the cross-covariance function and the non-stationarity of the multivariate output. We facilitate the computational complexity of the Markov chain Monte Carlo (MCMC) sampler by choosing appropriately the covariance function and prior distributions. Based on the BTMGP, we develop a sequential design of experiment for the input space and construct an emulator. We demonstrate the use of the proposed method in test cases and compare it with alternative approaches. We also apply the sequential sampling technique and BTMGP to model the multiphase flow in a full scale regenerator of a carbon capture unit.

Keywords: multivariate Gaussian process, separability, Bayesian treed Gaussian process, Markov chain Monte Carlo, computer experiments

1 Introduction

Using numerical simulations to model the behavior of large-scale complex systems is common in many fields of science and technology. While improving the level of detail and model resolution may increase the accuracy of these simulations compared to real systems, the increase in the associated computation cost may be significant. Despite the availability of faster and parallelized computational resources, it often is too expensive to run such complex models for all possible input conditions.

One example of expensive and complex models is computational fluid dynamics (CFD)-based simulations of a post-combustion carbon capture device (details in Section 6.1). Carbon

*Corresponding authors gratefully acknowledge

capture is an alternative approach to limit greenhouse gas emissions from thermal power plants (MacDowell et al., 2010). Carbon capture simulations should include fluid flow dynamics, reaction kinetics, and heat transfer occurring in the system. Moreover, commercial devices can be as large as tens of meters in height, requiring a large number of computational cells to obtain reasonably accurate solutions. For such large-scale systems, each simulation may take several days or even weeks to run. Parametric studies for varying operating conditions and material properties, such as those performed in Sarkar et al. (2014), become infeasible when reaction kinetics and heat transfer are included. The simulations become even more expensive computationally when these additional physical complexities are introduced. Therefore, time efficient surrogate models derived from a finite number of simulations need to be developed.

In this paper, we are interested in emulating the flow inside the regenerator of a carbon capture unit. The quantity of interest is the density function describing the distribution of volume fractions of sorbent particles in the device. The sorbent, comprised of small chemically reactive particles flowing through the device, is capable of reacting with the carbon dioxide and removing it from the thermal power plant exhaust. The distribution of the sorbent particles inside the regenerator, predicted by the CFD model, strongly affects carbon capture efficiency. Therefore, in a local region inside the regenerator, the relative volume occupied by sorbent particles, i.e., the sorbent volume fraction, is of interest. Using fewer CFD simulations, a surrogate model for predicting the distribution function of sorbent volume fraction will be developed, along with uncertainty estimates for the predictions.

Several methods based on the Gaussian process (GP) (Cressie, 1993) have been proposed to build surrogate models used to predict the response surface with only a few observations. Often, these methods make it possible to emulate the simulator output to a high degree of precision using only a few hundred runs, or fewer, of the simulator. Typically, the GP is used successfully as an emulator because it can investigate and incorporate dependencies involved with multivariate output, e.g., (Mardia and Goodall, 1993; Conti and O'Hagan, 2010). These works have concentrated on the stationary multivariate Gaussian process (MGP) but not much attention has been given to computer experiments with non-stationary output. We refer to a MGP as as non-stationary if its mean has abrupt changes or its behavior depends on their actual position and not on the distance between any two realizations.

Partitioning provides a straightforward mechanism for creating a non-stationary model and can help ease computational demands by fitting models to less data. Kim et al. (2005) used the Voronoi tessellation to partition the space into an independent stationary GP. Denison et al. (1998) and Chipman et al. (1998) used a Bayesian tree with different priors to partition the input space into simple linear regressions. Gramacy and Lee (2008) generalized, with the

Bayesian treed Gaussian process (BTGP), the Bayesian tree proposed by Chipman et al. (1998) to partition the input space into multiple stationary GPs. Despite the success of the BTGP, it has only been developed to analyze each component of the multivariate output separately.

Sampling methods have been developed to increase accuracy of the emulator and improve predictability. Proposed methods, such as Latin hypercube sampling (LHS) (Iman and Conover, 1980), orthogonal arrays, and multilevel Monte Carlo (Giles, 2008), have been developed to sample the input space without considering information about the output. Other more sophisticated methods use the active learning sequential design of experiment, such as Active Learning MacKay (ALM) and Active Learning Cohn (ALC) (Seo et al., 2000; Gramacy and Lee, 2009). These methods depend on what we already know about the output and are based on the predictive variance of the candidate input samples.

In this paper, we develop a novel Bayesian treed multivariate Gaussian process (BTMGP) to model the uncertainty of multivariate and non-stationary computer experiment output. The input domain is partitioned into disjoint subregions with a binary tree similar to the classification and regression tree (CART) (Chipman et al., 1998) and BTGP (Gramacy and Lee, 2008). Each subregion, which corresponds to one external node of the tree, is modeled independently by a MGP with separable covariance function as in Conti and O'Hagan (2010). The Kronecker product of the separable model simplifies the computations in each external node. We design local moves for the reversible jump Markov chain Monte Carlo (MCMC) involved in the Bayesian tree operations that lead to a satisfactory mixing of the MCMC sampler. The BTMGP can be used to explicitly predict the non-stationary response surface and represent the uncertainty associated with it. In addition, we extend the univariate ALC technique to sequentially sample multivariate output from the most informative input with the help of the proposed BTMGP and expert prior knowledge. In a Bayesian formulation, a prior distribution of the computer experiment output is updated in light of new observations and is conditional on various hyperparameters. By using an active learning sequential design of experiment, we can determine the response surface of the multivariate output from a simulator using well-selected observations. The simultaneous BTMGP fitting and sequential adaptive sampling provide a good prediction fit even for a small number of realizations. Finally, we apply our method to computer experiments of a regenerator device from a carbon capture plant.

The rest of the paper is organized as follows: in Section 2, we review the statistical models, and in Section 3 we describe the Bayesian inference and prediction. In Section 4, we introduce an adaptive sampling technique. We conduct an artificial example study in Section 5, while we use our proposed method to analyze the multiphase flow in a full-scale regenerator of the carbon capture problem in Section 6. Conclusions are presented in Section 7.

2 Statistical model

Consider a computer model with input domain $\mathcal{X} \subset \mathbb{R}^l$, where l is the dimension of the input space. Also, let $\mathbf{f}(\boldsymbol{x}_i) \in \mathbb{R}^q$ denote the $1 \times q$ vector observed output at input $\boldsymbol{x}_i$; $\tilde{\boldsymbol{Y}} = (\mathbf{f}(\boldsymbol{x}_1), \ldots, \mathbf{f}(\boldsymbol{x}_n))^T$ denote the $(nq) \times 1$ observed output vector; $\boldsymbol{Y} = (\mathbf{f}^T(\boldsymbol{x}_1), \ldots, \mathbf{f}^T(\boldsymbol{x}_n))^T$ denote the $n \times q$ observed output matrix; and $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T$ represent the $n \times l$ observed input matrix. The challenge in the multivariate process is to model its parameters in the presence of non-stationarity. In this paper, we used Bayesian treed ideas coupled with MGP with separable covariance function. In the following we present two of the main components of the proposed method: the MGP with separable covariance function and the Bayesian tree.

2.1 Gaussian process with separable covariance

The basic multivariate Gaussian regression model often used in geostatistics, is of the form:

$$\mathbf{f}(\boldsymbol{x}) = \boldsymbol{h}^T(\boldsymbol{x})\boldsymbol{B} + \boldsymbol{w}(\boldsymbol{x}) + \boldsymbol{\epsilon}(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{h}(\boldsymbol{x})$ is the $m \times 1$ vector of basis functions at $\boldsymbol{x}$, $\boldsymbol{B}$ is the linear regression coefficient of dimension $m \times q$, $\boldsymbol{w}(\boldsymbol{x})$ is a zero mean MGP, and $\boldsymbol{\epsilon}(\boldsymbol{x})$ is the nugget error.

The separable model, a special case for the covariance function of $\boldsymbol{w}(\boldsymbol{x})$, frequently has been used in previous works (Mardia and Goodall, 1993; Conti and O'Hagan, 2010). The covariance function can be written as $c(\boldsymbol{w}(\boldsymbol{x}), \boldsymbol{w}(\boldsymbol{x}')) = \rho(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\psi})\boldsymbol{\Sigma}$, where $\boldsymbol{\Sigma}$ is the $q \times q$ variance matrix of $\mathbf{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_q(\boldsymbol{x}))\boldsymbol{h}^T(\boldsymbol{x})$ at any location $\boldsymbol{x}$, $\rho(.,.;\boldsymbol{\psi})$ is a known correlation function (e.g., the power exponential, rational quadratic, and Matérn), and $\boldsymbol{\psi}$ are the parameters associated with the correlation function of the input. This form of the covariance function assumes the same correlation parameters for every $f_i(\boldsymbol{x})$. For computational purposes, the nugget error $\epsilon(\boldsymbol{x})$ may be assumed equal to zero.

The covariance matrix of the vector $\tilde{\boldsymbol{Y}}$ can be written as $\boldsymbol{C} = \boldsymbol{R} \otimes \boldsymbol{\Sigma}$, where $\boldsymbol{R} \in \mathbb{R}^{n \times n}$ is the correlation matrix generated by $\boldsymbol{X}$ and $\rho(\,\cdot\,,\,\cdot\,;\boldsymbol{\psi})$ ($\boldsymbol{R}(i,j) = [\rho(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\psi})]$ is the correlation of the $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ input). This representation of the covariance matrix facilitates the computation of the likelihood, which depends on the determinant and inverse of $\boldsymbol{C}$. The determinant can be expressed as $|\boldsymbol{C}| = |\boldsymbol{R}|^q|\boldsymbol{\Sigma}|^n$ and the inverse as $\boldsymbol{C}^{-1} = \boldsymbol{R}^{-1} \otimes \boldsymbol{\Sigma}^{-1}$. The likelihood can be written as a function of the matrix $\boldsymbol{Y}$:

$$\log L(\boldsymbol{Y};\,\cdot\,) = const - \frac{1}{2}\log(|\boldsymbol{R}|^q|\boldsymbol{\Sigma}|^n|) - \frac{1}{2}\operatorname{tr}(\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{HB})^T\boldsymbol{R}^{-1}(\boldsymbol{Y} - \boldsymbol{HB})), \tag{2}$$

where $\boldsymbol{Y}$ is a $n \times q$ matrix and $\boldsymbol{HB}$ is the multivariate linear regression mean of $\boldsymbol{Y}$.

The preceding model requires the specification of the correlation functions ρ. The com-

putational simplification of the Kronecker product requires the nugget error to be zero. As a result, we, in practice, we may experience computation instabilities. A remedy for the ill-conditioned matrix is to assume that in every separate correlation matrix, a positive quantity exists in the diagonals similar to Bilionis et al. (2013). We assume a nugget random parameter for the correlation function that has to be estimated. The modified correlation function is assumed $\rho(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\psi}) = \tilde{\rho}(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\lambda}) + \boldsymbol{g}^2 \delta_{x,x'}$,where $\boldsymbol{\psi} = (\boldsymbol{\lambda}, \boldsymbol{g})$, $\lambda = (\lambda_1, \ldots, \lambda_l)$, where λ_k represents the correlation strength of $\boldsymbol{w}(\cdot)$ in the k^{th} direction, and $\boldsymbol{g}$ is the nugget quantity used for the stability of the input correlation matrix. In this paper, we use the square exponential $\tilde{\rho}(\boldsymbol{\xi}, \boldsymbol{\xi}'; \boldsymbol{\lambda}_x) = \exp\{-\frac{1}{2}\sum_{k=1}^{l}(x_k - x'_k)^2/\lambda_k^2\}$, which has been proven to work well in previous studies. Despite $\rho(\boldsymbol{x}, \boldsymbol{x}; \boldsymbol{\psi}) > 1$ caused by the positive nugget term $\boldsymbol{g}$, we call $\rho(\cdot)$ correlation function for notational convenience.

2.2 Bayesian tree

The separable model can be too simplistic to deal with computer experiments in practice. In many applications, a stationary MGP may not be appropriate due to the non-stationary structure of the data. The mean, variance, and spatial dependency may differ from one input subregion to another.

A Bayesian tree (Chipman et al., 1998; Gramacy and Lee, 2008) partitions the input space into non-overlapping regions by making binary splits recursively. Chipman et al. (1998) employ a piecewise constant and Gramacy and Lee (2008) use a GP inside of each external node. Each new partition is a sub-partition of a previous one. Conditional on a treed partition, the prediction from the BTGP model is done independently within each subregion, following the conventional GP prediction technique (Hjort and Omre, 1994). Although each realization of the predictive surface is discontinuous, the aggregated posterior predictive mean surface tends to produce approximately smooth transitions between subregions. To extend the described Bayesian treed model to multivariate output, we follow the same setting.

In the Bayesian framework, a binary tree is treated as random and assigned with a prior distribution via a tree-generating process (Chipman et al., 1998). Starting with a null tree (all data in a single region), a leaf node $\eta \in \mathcal{T}$, representing a subregion of the input space, splits with probability $P_{\text{split}}(\eta, \mathcal{T}) = a(1 + d_\eta)^{-b}$, where d_η is the depth of $\eta \in \mathcal{T}$, a controls the balance of tree's shape, and b controls the tree's size.

The treed model prior is:

$$P(\mathcal{T}) = P_{rule}(\rho|\eta, \mathcal{T}) \prod_{\eta_i \in \mathcal{I}} P_{split}(\eta_i, \mathcal{T}) \prod_{\eta_j \in \mathcal{D}} (1 - P_{split}(\eta_j, \mathcal{T})),$$

where $\mathcal{I}$ and $\mathcal{D}$ denote the internal and terminal nodes, respectively. $P_{rule}(\rho|\eta, \mathcal{T})$ involves

the splitting process, which initially chooses the splitting dimension u from a discrete uniform distribution. Then, the split location ζ is chosen uniformly from a subset of the locations S in the u^{th} dimension.

3 Bayesian inference

We consider a partition $\{\mathcal{X}_1, \ldots, \mathcal{X}_D\}$ of disjoint subregions of the input domain $\mathcal{X}$, such that $\mathcal{X} = \bigcup_{i=1}^{D} \mathcal{X}_i$, corresponds to a tree structure $\mathcal{T}$ with D external nodes. We model each partition $\{\mathcal{X}_i\}_{i=1}^{D}$ with a MGP with likelihood $L_i(\boldsymbol{Y}_i|\boldsymbol{\phi}_i)$ as defined in Section 2.1, where $\boldsymbol{\phi}_i = (\boldsymbol{B}_i, \boldsymbol{\lambda}_i, \boldsymbol{g}_i, \boldsymbol{\Sigma}_i)$ denotes the parameters of the MGP of the i^{th} external node. The joint likelihood defined on the whole domain $\mathcal{X}$ is:

$$L(\boldsymbol{Y}|\boldsymbol{\phi}, \mathcal{T}) = \prod_{j=1}^{n} \sum_{i=1}^{D} L_i(\mathbf{f}(\boldsymbol{x}_j)|\boldsymbol{\phi}_i) \mathbf{1}_{\{\mathcal{X}_i\}}(\boldsymbol{x}_j),$$

where $\boldsymbol{\phi} = (\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_D)$ and $\mathbf{1}_{\{\mathcal{X}_i\}}(\boldsymbol{x}_j)$ is the indicator function, which is equal to 1 if $\boldsymbol{x}_j \in \mathcal{X}_i$ and 0 otherwise.

According to the Bayesian framework, we assign a prior distribution on the parameter $(\mathcal{T}, \boldsymbol{\phi})$, such as $\pi(\mathcal{T}, \boldsymbol{\phi}) = \pi(\mathcal{T}) \prod_{i=1:D} \pi(\boldsymbol{B}_i) p(\boldsymbol{\Sigma}_i) p(\boldsymbol{\lambda}_i) p(\boldsymbol{g}_i)$. Here, we consider that the MGP parameters $(\boldsymbol{B}_i, \boldsymbol{\lambda}_i, \boldsymbol{g}_i, \boldsymbol{\Sigma}_i)$ are apriori independent between different partitions and independent of each other within the partitions of the input domain. However, the proposed method is not limited solely to this prior model.

The marginal prior distribution of the binary tree $\pi(\mathcal{T})$ is defined according to the tree generating process suggested by Chipman et al. (1998) and discussed in Section 2.2. Moreover, for $\boldsymbol{\phi}$, we choose the same prior specification of the local MGP parameters for each of the terminal nodes. Given a tree partition $\mathcal{X}_i$ (that corresponds to a terminal node η_i of the binary tree), we assign conjugate prior distributions on the $(\boldsymbol{B}_i, \boldsymbol{\Sigma}_i)$ to obtain a more convenient form for the posterior distributions. More precisely, for the mean coefficient $\boldsymbol{B}_i$, we consider a multivariate normal prior distribution $\mathcal{N}(0, \boldsymbol{I}\sigma^2_{\boldsymbol{B}})$ with mean zero and variance $\boldsymbol{I}\sigma^2_{\boldsymbol{B}}$. The prior distribution on $\boldsymbol{\Sigma}_i$ is an inverse Wishart, $IW(r, \boldsymbol{\Omega})$. We consider non-informative prior $\pi(\boldsymbol{B}_i, \boldsymbol{\Sigma}_i) \propto |\boldsymbol{\Sigma}_i|^{-\frac{q+1}{2}}$, which is a limiting case for $\Omega \to 0$ and $\sigma_{\boldsymbol{B}} \to \infty$. This will lead to a closed form of the marginal posterior distribution of $\boldsymbol{\lambda}_i$ (Conti and O'Hagan, 2010).

The prior of $\lambda_{i,j}$, for $i = 1, \ldots, D$ and $j = 1, \ldots, l$, is more complicated and may depend on the correlation function used. For example, for exponential correlation function a non-informative distribution for $\lambda_{i,j}$ is $IG(2, b_0)$ where $b_0 = x_0/(-2\ln(0.05))$ and x_0 is the maximum distance of the i^{th} direction (Banerjee et al., 2004). In this paper, we choose the mixture of Gamma distributions similar to those of Gramacy and Lee (2008). Finally, we assign an

exponential distribution prior for the nugget standard deviation $\boldsymbol{g}_i$.

Because the resulting posterior distribution is intractable, we use MCMC methods to carry out inference. A blockwise MCMC sampler (Gelfand and Smith, 1990) is used to simulate each component of $\mathcal{T}|\boldsymbol{\phi}$ and $\boldsymbol{\phi}|\mathcal{T}$.

3.1 Within-tree MCMC simulation

Given a fixed tree structure $\mathcal{T}$, the parameters $\boldsymbol{\phi}_i$ of the MGP in each of the external nodes ($i = 1, \ldots, D$) are updated independently. The conditional distribution of $(\boldsymbol{B}_i, \boldsymbol{\Sigma}_i)$ can be factorized as $p(\boldsymbol{B}_i, \boldsymbol{\Sigma}_i|\boldsymbol{Y}_i, \boldsymbol{\lambda}_i, \boldsymbol{g}_i) = p(\boldsymbol{B}_i|\boldsymbol{Y}_i, \boldsymbol{\Sigma}_i, \boldsymbol{\lambda}_i, \boldsymbol{g}_i)p(\boldsymbol{\Sigma}_i|\boldsymbol{Y}_i, \boldsymbol{\lambda}_i, \boldsymbol{g}_i)$, where:

$$\begin{aligned}\boldsymbol{B}_i|\boldsymbol{Y}_i, \boldsymbol{\Sigma}_i, \boldsymbol{\lambda}_i, \boldsymbol{g}_i \sim& \mathcal{N}_{m,q}(\hat{\boldsymbol{B}}_i, (\boldsymbol{H}_i^T \boldsymbol{R}_i^{-1} \boldsymbol{H}_i), \boldsymbol{\Sigma}_i),\\ \boldsymbol{\Sigma}_i|\boldsymbol{Y}_i, \boldsymbol{\lambda}_i, \boldsymbol{g}_i \sim& IW((r + n_i), (n_i - m)\hat{\boldsymbol{\Sigma}}_i),\end{aligned}$$

with $\hat{\boldsymbol{B}}_i = (\boldsymbol{H}_i^T \boldsymbol{R}_i^{-1} \boldsymbol{H}_i)^{-1} \boldsymbol{H}_i^T \boldsymbol{R}_i^{-1} \boldsymbol{Y}_i$ and $\hat{\boldsymbol{\Sigma}}_i = (n_i - m)^{-1}(\boldsymbol{Y}_i - \boldsymbol{H}_i\hat{\boldsymbol{B}}_i)^T \boldsymbol{R}_i^{-1}(\boldsymbol{Y}_i - \boldsymbol{H}_i\hat{\boldsymbol{B}}_i)$.

For $i = 1, \ldots, D$, full conditional posterior distribution of $\boldsymbol{\lambda}_i, \boldsymbol{g}_i|\boldsymbol{Y}, \Sigma_i, \boldsymbol{B}_i$ for an arbitrary choice of $\pi(\boldsymbol{\lambda}_i, \boldsymbol{g}_i)$ is such that:

$$p(\boldsymbol{\lambda}_i, \boldsymbol{g}_i|\boldsymbol{Y}_i, \boldsymbol{\Sigma}_i, \boldsymbol{B}_i) \propto \pi(\boldsymbol{\lambda}_i, \boldsymbol{g}_i)|\boldsymbol{R}_i|^{-\frac{q}{2}} \exp(-\frac{1}{2}\operatorname{tr}(\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{Y}_i - \boldsymbol{H}_i\boldsymbol{B}_i)^T \boldsymbol{R}_i^{-1}(\boldsymbol{Y}_i - \boldsymbol{H}_i\boldsymbol{B}_i))). \quad (3)$$

For $i = 1, \ldots, D$, conditional posteriors of $\boldsymbol{\lambda}_i$ and $\boldsymbol{g}_i$ cannot be sampled directly. Therefore, we use Metropolis-Hastings updates within a Gibbs sampler, (Mueller, 1993; Gelfand and Smith, 1990; Hastings, 1970). Given the prior specification for $\boldsymbol{B}_i$ and $\boldsymbol{\Sigma}_i$ in the previous section and integrating out $\boldsymbol{B}_i$ and $\boldsymbol{\Sigma}_i$ from the posterior of $\boldsymbol{\lambda}_i, \boldsymbol{g}_i, \boldsymbol{\Sigma}_i, \boldsymbol{B}_i|\boldsymbol{Y}_i$, the conditional distribution of $\boldsymbol{\lambda}_i, \boldsymbol{g}_i|\boldsymbol{Y}$ can be expressed similarly to Conti and O'Hagan (2010) as:

$$p(\boldsymbol{\lambda}_i, \boldsymbol{g}_i|\boldsymbol{Y}_i) \propto \pi(\boldsymbol{g}_i)\pi(\boldsymbol{\lambda}_i)|\boldsymbol{R}_i|^{-\frac{q}{2}}|\boldsymbol{H}_i^T \boldsymbol{R}_i^{-1} \boldsymbol{H}_i|^{-\frac{q}{2}}(\hat{\boldsymbol{\Sigma}}_i)^{\frac{n_i - m}{2}}, \quad (4)$$

where m is the total number of basis functions in the model and $\hat{\boldsymbol{\Sigma}}_i$ is the generalized least square estimator of $\boldsymbol{\Sigma}_i$ given above. This representation is appealing because it simplifies the problem into separate matrices. Yet, integrating equation (3) over $\boldsymbol{B}_i$ and $\boldsymbol{\Sigma}_i$ can improve the mixing of the MCMC (Liu et al., 1994; Berger et al., 2001). For every external node we impose the constrain $n_i > m + q$, such that the posteriors of all parameters are proper, before initiating sampling from the posterior distribution. Moreover, given this restriction, it is possible to integrate out both $\boldsymbol{B}_i$ and $\boldsymbol{\Sigma}_i$, resulting in the predictive distribution of $\mathbf{f}(\boldsymbol{x})$ conditional only on $\boldsymbol{\lambda}_i, \boldsymbol{g}_i$ (as shown in Section 3.3). This constraint must be maintained when generating the Bayesian tree.

3.2 Across-tree MCMC simulation

The structure of the binary tree of the MGP model is updated through a random scan MCMC sweep that includes (as updates) the *Change*, *Swap*, *Rotate*, and *Grow & Prune* operations. These operations are similar to the BTGP operations introduced by Gramacy and Lee (2008) with differences in the covariance structure and, thus, the likelihood. The first three operations are Metropolis-Hastings updates operating on fixed dimensional spaces, while the last two are a reversible jump pair of moves (Green, 1995) that perform changes to the dimension of the parameter space. For more details, refer to Chipman et al. (1998) and Gramacy and Lee (2008).

The *Grow & Prune* operations are a reversible jump pair of moves that change the structure of the binary tree by adding or removing nodes while they change the dimension of the parametric space. Given the current state is at binary tree $\mathcal{T}$, the *Grow* operation involves several steps. We randomly select an external node η_{j_0} that corresponds to a subregion $\mathcal{X}_{j_0}$ with data $\{\boldsymbol{X}_{j_0}, \boldsymbol{Y}_{j_0}\}$ and MGP model with parameters $\boldsymbol{\phi}_{j_0} = (\boldsymbol{B}_{j_0}, \boldsymbol{\lambda}_{j_0}, \boldsymbol{g}_{j_0}, \boldsymbol{\Sigma}_{j_0})$. We propose node η_{j_0} to split into two new child nodes η_{j_1} and η_{j_2} according to the splitting rule P_{rule} used in the priors, and we denote the proposed tree as $\mathcal{T}'$. We consider that nodes η_{j_1} and η_{j_2} correspond to disjoint subregions $\mathcal{X}_{j_1}$ and $\mathcal{X}_{j_2}$, the union of which is $\mathcal{X}_{j_0}$, with data $\{\boldsymbol{X}_{j_1}, \boldsymbol{Y}_{j_1}\}$ and $\{\boldsymbol{X}_{j_2}, \boldsymbol{Y}_{j_2}\}$, respectively. Let $\boldsymbol{\phi}_{j_1} = (\boldsymbol{B}_{j_1}, \boldsymbol{\lambda}_{j_1}, \boldsymbol{g}_{j_1}, \boldsymbol{\Sigma}_{j_1})$ and $\boldsymbol{\phi}_{j_2} = (\boldsymbol{B}_{j_2}, \boldsymbol{\lambda}_{j_2}, \boldsymbol{g}_{j_2}, \boldsymbol{\Sigma}_{j_2})$ denote the parameter vectors of the MGP associated with the new nodes η_{j_1} and η_{j_2}. A newly formed child, lets say η_{j_1}, is randomly chosen to receive values for $(\boldsymbol{\lambda}_{j_1}, \boldsymbol{g}_{j_1})$ from the parent such that $(\boldsymbol{\lambda}_{j_1}, \boldsymbol{g}_{j_1}) = (\boldsymbol{\lambda}_{j_0}, \boldsymbol{g}_{j_0})$. Meanwhile, for the other, $(\boldsymbol{\lambda}_{j_2}, \boldsymbol{g}_{j_2})$, we generate values from a proposal $Q(\boldsymbol{\lambda}_{j_2}, \boldsymbol{g}_{j_2})$. $Q(\boldsymbol{\lambda}_{j_2}, \boldsymbol{g}_{j_2})$ can be the prior distribution of $(\boldsymbol{\lambda}_{j_2}, \boldsymbol{g}_{j_2})$. We generate proposals for $(\boldsymbol{B}_{j_1}, \boldsymbol{\Sigma}_{j_1})$ and $(\boldsymbol{B}_{j_2}, \boldsymbol{\Sigma}_{j_2})$ from the posterior conditional distributions $p(\boldsymbol{B}_{j_1}, \boldsymbol{\Sigma}_{j_1} | \boldsymbol{Y}_{j_1}, \boldsymbol{\lambda}_{j_1}, \boldsymbol{g}_{j_1})$ and $p(\boldsymbol{B}_{j_2}, \boldsymbol{\Sigma}_{j_2} | \boldsymbol{Y}_{j_2}, \boldsymbol{\lambda}_{j_2}, \boldsymbol{g}_{j_2})$. Let G and P' denote the set of the growable nodes of $\mathcal{T}$ and prunable nodes of $\mathcal{T}'$, respectively. The *Grow* operation is accepted with probability $\min\{1, A\}$, where

$$A = \frac{1 - a(1 + d_{\eta_{j_0}})^{-b}}{a(1 + d_{\eta_{j_0}})^{-b}(1 - a(2 + d_{\eta_{j_0}})^{-b})^2} \frac{|G|}{|P'|} \frac{p(\boldsymbol{\lambda}_{j_1}, \boldsymbol{g}_{j_1} | \boldsymbol{Y}_{j_1}) p(\boldsymbol{\lambda}_{j_2}, \boldsymbol{g}_{j_2} | \boldsymbol{Y}_{j_2})}{p(\boldsymbol{\lambda}_{j_0}, \boldsymbol{g}_{j_0} | \boldsymbol{Y}_{j_0}) Q(\boldsymbol{\lambda}_{j_2}, \boldsymbol{g}_{j_2})}. \tag{5}$$

The *Prune* operation is the reverse analog *of Grow*, from tree $\mathcal{T}'$ to $\mathcal{T}$, and designed so the detailed balanced condition is satisfied. The operation is accepted with probability $\min\{1, 1/A\}$.

Remark 1: Note that the *Grow/ Prune* and *Change* operations must be accepted upon the condition $n_j > m + q$. Each of these operations must satisfy the constraints of the separable model to ensure a proper posterior.

Remark 2: For the multivariate case, Gramacy and Lee (2008) separately implement q different univariate Bayesian trees using parallel computing. Basically, their method can be considered as a one-by-one, separate univariate analysis for each of the outputs. The proposed

BTMGP model uses only one tree and a covariance function that can model the dependence between random variables. For one-dimensional output, our model is the same, in principle, as the BTGP in Gramacy and Lee (2008), with some differences in the formulation.

Remark 3: In practice, it is unnecessary to generate proposal values or compute the proposal densities for $(\boldsymbol{B}_{j_1}, \boldsymbol{\Sigma}_{j_1})$ and $(\boldsymbol{B}_{j_2}, \boldsymbol{\Sigma}_{j_2})$ (or $(\boldsymbol{B}_j, \boldsymbol{\Sigma}_{j_0})$) before a *Grow* (or *Prune*) operation is accepted or rejected. This is because these parameters are not involved in the computation of the acceptance ratio A (or $1/A$) or the generation of other proposals within *Grow/ Prune* operations. However, after a *Grow* (or *Prune*) operation has been accepted, $(\boldsymbol{B}_{j_1}, \boldsymbol{\Sigma}_{j_1})$ and $(\boldsymbol{B}_{j_2}, \boldsymbol{\Sigma}_{j_2})$ (or $(\boldsymbol{B}_{j_0}, \boldsymbol{\Sigma}_{j_0})$) can be generated from the conditional posterior distributions if inference on these parameters is of interest.

Remark 4: The fact that we are able to use the conditional posterior distributions of the linear coefficient and covariance matrices as reversible jump proposals is possible to lead to more acceptable *Grow/ Prune* operations, as discussed by Karagiannis and Andrieu (2013) and Godsill (2001), and creates simpler acceptance ratios at relatively low computational cost. This is particularly important here given the multidimensional nature of the model.

3.3 Predictive distribution

In this section, we calculate the predictive distribution of $\boldsymbol{y}(\boldsymbol{x}') = \mathbf{f}(\boldsymbol{x}') \in \mathbb{R}^q$ at a new input point, $\boldsymbol{x}' \in \mathbb{R}^{k_x}$. The predictive distribution can be used to predict the response surface and its associated error.

Given the data, the tree, and the parameters of the MGP in each external leaf, the conditional posterior distribution for the emulator $\mathbf{f}(\,\cdot\,)$ is:

$$p(\mathbf{f}(\boldsymbol{x}')|\boldsymbol{B}, \boldsymbol{\Sigma}, \boldsymbol{g}, \lambda, \mathcal{T}, \boldsymbol{Y}) \equiv \sum_{i=1:D} \mathbf{1}_{\{\mathcal{X}_i\}}(\boldsymbol{x}') \mathcal{N}_q \left(\boldsymbol{m}_i^*(\boldsymbol{x}'; \boldsymbol{B}_i), r_i^*(\boldsymbol{x}', \boldsymbol{X}_i; \boldsymbol{g}_i, \lambda_i) \boldsymbol{\Sigma}_i \right), \tag{6}$$

where $\boldsymbol{m}_i^*(\boldsymbol{x}'; \boldsymbol{B}_i) = \boldsymbol{B}_i^T \boldsymbol{h}_i(\boldsymbol{x}') + \boldsymbol{r}(\boldsymbol{X}_i, \boldsymbol{x})^T \boldsymbol{R}_i^{-1} (\boldsymbol{Y}_i - \boldsymbol{H}_i \boldsymbol{B}_i)$ and $r_i^*(\boldsymbol{x}', \boldsymbol{X}_i; \boldsymbol{g}_i, \lambda_i) = (\boldsymbol{r}_i(\boldsymbol{x}', \boldsymbol{x}') - \boldsymbol{r}(\boldsymbol{X}_i, \boldsymbol{x})^T \boldsymbol{R}_i^{-1} \boldsymbol{r}(\boldsymbol{X}_i, \boldsymbol{x}))$. The preceding representation can be further simplified using the conditional distribution of $\mathbf{f}(\,\cdot\,)|\boldsymbol{Y}, \boldsymbol{g}, \lambda$. Given the prior specification of $\pi(\boldsymbol{B}_i, \boldsymbol{\Sigma}_i) \propto |\boldsymbol{\Sigma}_i|^{-\frac{q+1}{2}}$ and integrating out $\boldsymbol{B}_i$ and $\boldsymbol{\Sigma}_i$, the distribution of $\mathbf{f}(\boldsymbol{x}')|\mathcal{T}, \boldsymbol{g}, \lambda, \boldsymbol{Y} \equiv \mathbf{f}(\boldsymbol{x}')|\boldsymbol{g}_i, \lambda_i, \boldsymbol{Y}_i$, if $\boldsymbol{x}' \in \mathcal{X}_i$, is a multivariate $\boldsymbol{t}$-student (Conti and O'Hagan, 2010):

$$p(\mathbf{f}(\boldsymbol{x}')|\mathcal{T}, \boldsymbol{g}, \lambda, \boldsymbol{Y}) \equiv \sum_{i=1:D} \mathbf{1}_{\{\mathcal{X}_i\}}(\boldsymbol{x}') \mathcal{T}_q \left(\boldsymbol{m}_i^*(\boldsymbol{x}'; \hat{\boldsymbol{B}}_i), r^*(\boldsymbol{x}', \boldsymbol{X}_i; \boldsymbol{g}_i, \lambda_i) \hat{\boldsymbol{\Sigma}}_i; n_i - m \right), \tag{7}$$

with $n_i - m$ representing the degrees of freedom of the $\boldsymbol{t}$-distribution.

The Bayesian predictive density function $\mathbf{f}(\,\cdot\,)|\boldsymbol{Y}$ is calculated through Bayesian model av-

eraging (BMA) as

$$p(\mathbf{f}(\boldsymbol{x})|\boldsymbol{Y}) = \sum_{\mathcal{T}} \int_{\boldsymbol{\lambda},\boldsymbol{g}} p(\mathbf{f}(\boldsymbol{x})|\boldsymbol{\lambda},\boldsymbol{g},\boldsymbol{Y},\mathcal{T})\pi(\boldsymbol{\lambda},\boldsymbol{g},\mathcal{T}|\boldsymbol{Y})d\boldsymbol{\lambda}d\boldsymbol{g}. \tag{8}$$

In practice, exhausting enumeration and summation over all possible $\mathcal{T}$ in equation (8) is not feasible. Moreover, the integral in equation (8) is computationally intractable. Thus, the following numerical methods are required for the evaluation of the predictive density function $p(\mathbf{f}(\boldsymbol{x})|\boldsymbol{Y})$:

1. Generate MCMC samples $(\boldsymbol{\lambda}^{(1)},\boldsymbol{g}^{(1)},\mathcal{T}^{(1)}),\ldots,(\boldsymbol{\lambda}^{(M)},\boldsymbol{g}^{(M)},\mathcal{T}^{(M)})$ from $p(\boldsymbol{\lambda},\boldsymbol{g}|\boldsymbol{Y})$ as we described in Section 3.2 and 3.1.

2. Approximate $p(\mathbf{f}(\boldsymbol{x})|\boldsymbol{Y})$ by $\hat{p}(\mathbf{f}(\boldsymbol{x})|\boldsymbol{Y}) = 1/M \sum_{k=1}^{M} p(\mathbf{f}(\boldsymbol{x})|\mathcal{T}^{(k)},\boldsymbol{\lambda}^{(k)},\boldsymbol{g}^{(k)},\boldsymbol{Y})$.

This predictive process tends to smooth the prediction surface around the tree limit regions edges (refer to Gramacy and Lee (2008)). Also, the proposed method allows the computation of the predictive distribution of any function of $\mathbf{f}$ without relying on approximation transformation methods, such as the delta method. As we observe in our application section, this is crucial.

4 Sequential design of experiments via active learning

The *sequential design of experiments* (SDOE) via *active learning* is a popular sequential data collection method in computer experiments (Gramacy and Lee, 2009). The main goal of SDOE is to select a subset of points on the input space that maximizes a utility function chosen by the practitioner. *Active Learning MacKay* (ALM) and *Active Learning Cohn* (ALC) are the two main approaches for SDOE via active learning. The ALM approach uses the concept of maximizing the information gained by sequentially selecting a subset of the input point, which has the greatest uncertainty in the output space. On the other hand, the ALC approach (Cohn, 1996) sequentially selects a subset of data by maximizing the expected reduction in mean squared error averaged over the whole input space:

$$\Delta\hat{\sigma}^2(\tilde{\boldsymbol{x}}) = \int_{\mathcal{X}} \Delta\hat{\sigma}^2_{\tilde{\boldsymbol{x}}}(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z} = \int_{\mathcal{X}} (\hat{\sigma}^2(\boldsymbol{z}) - \hat{\sigma}^2_{\tilde{x}}(\boldsymbol{z}))p(\boldsymbol{z})d\boldsymbol{z}, \tag{9}$$

where $\Delta\hat{\sigma}^2_{\tilde{\boldsymbol{x}}}(\boldsymbol{z})$ is the reduction of variance of the output in location $\boldsymbol{z}$ when an observation in location $\tilde{\boldsymbol{x}}$ is added. Also, $p(\boldsymbol{z})$ is the input variable density function, which can be considered as a prior of the input space (where a generalized Beta or truncated Normal distribution is a convenient choice in practice), $\sigma^2(\boldsymbol{z})$ is the variance of output in location $\boldsymbol{z}$ without observing the output in location $\tilde{\boldsymbol{x}}$, and $\hat{\sigma}^2_{\tilde{\boldsymbol{x}}}(\boldsymbol{z})$ is the variance at location $\boldsymbol{z}$ when an observation at location

$\tilde{\boldsymbol{x}}$ exists. The integral in equation (9) usually is analytically intractable. Therefore, we compute it numerically by choosing a predetermined subset of gridded input data. Seo et al. (2000) have shown empirically that ALC performs better than ALM but is computationally more expensive. A detailed description of these techniques for the univariate case can be found in Seo et al. (2000) and Gramacy and Lee (2009). In this work, we extend the univariate ALC to the multivariate case and couple it with the proposed BTMGP.

Let $\boldsymbol{X}^n$ denote the input points with n observations and $\boldsymbol{X}^{n+1} = [\boldsymbol{X}^n, \tilde{\boldsymbol{x}}]$. Also, let $\boldsymbol{R}^n \in \mathbb{R}^{n\times n}$ denote the correlation matrix generated by $\boldsymbol{X}^n$ and $\rho(\,\cdot\,,\,\cdot\,;\boldsymbol{\lambda},\boldsymbol{g})$, and $\boldsymbol{R}^{n+1} \in \mathbb{R}^{(n+1)\times(n+1)}$ denote the correlation matrix generated by $\boldsymbol{X}^{n+1}$ and $\rho(\,\cdot\,,\,\cdot\,;\boldsymbol{\lambda},\boldsymbol{g})$. We define a scalar quantity associated directly with the uncertainty pertaining the input location $\boldsymbol{z}$ as:

$$\hat{\sigma}^2(\boldsymbol{z}) = \mathrm{tr}\big\{(\boldsymbol{r}(\boldsymbol{z},\boldsymbol{z}) - \boldsymbol{r}(\boldsymbol{X}^n,\boldsymbol{z})^T(\boldsymbol{R}^n)^{-1}\boldsymbol{r}(\boldsymbol{X}^n,\boldsymbol{z}))\otimes\boldsymbol{\Sigma}\big\},$$

and

$$\hat{\sigma}^2_{\tilde{\boldsymbol{x}}}(\boldsymbol{z})) = \mathrm{tr}\big\{(\boldsymbol{r}(\boldsymbol{z},\boldsymbol{z}) - \boldsymbol{r}(\boldsymbol{X}^{n+1},\boldsymbol{z})^T(\boldsymbol{R}^{n+1})^{-1}\boldsymbol{r}(\boldsymbol{X}^{n+1},\boldsymbol{z}))\otimes\boldsymbol{\Sigma}\big\}.$$

This is the sum of the variances of all outputs at all different input points. This form simplifies the multivariate active learning technique into a univariate form that can be used to sample the input space.

Similar to Gramacy and Lee (2009), we sample points according to the multivariate ALC conditional on the BTMGP parameters $(\boldsymbol{\phi},\mathcal{T})$ at candidate locations $\tilde{\boldsymbol{X}}$. Using the treed Gaussian process solves both non-stationarity and computational complexity. Different helping intermediate steps, such as choosing a few good candidates (Gramacy and Lee, 2009), based on the maximum a posteriori (MAP) Bayesian tree obtained in the previous trial, also can be incorporated to speed up the ALC algorithm.

Gramacy and Lee (2009) also offer details on how to invert a covariance matrix at low cost in the ALC algorithm. We follow the same setting to invert the matrix $\boldsymbol{R}^{n+1}$ in terms of $\boldsymbol{R}^n$ by using:

$$\Delta\hat{\sigma}_{\tilde{\boldsymbol{x}}}(\boldsymbol{z}) = \frac{\mathrm{tr}(\Sigma)[\boldsymbol{r}(\boldsymbol{X}^n,\boldsymbol{z})^T(\boldsymbol{R}^n)^{-1}\boldsymbol{r}(\boldsymbol{X}^n,\tilde{\boldsymbol{x}}) - \boldsymbol{r}(\tilde{\boldsymbol{x}},\boldsymbol{z})]}{\boldsymbol{r}(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{x}}) - \boldsymbol{r}(\boldsymbol{X}^n,\tilde{\boldsymbol{x}})^T(\boldsymbol{R}^n)^{-1}\boldsymbol{r}(\boldsymbol{X}^n,\tilde{\boldsymbol{x}})}, \tag{10}$$

where the subscript n denotes the sample size. Because of the independence assumption of the Bayesian tree, if $\boldsymbol{z}$ and $\tilde{\boldsymbol{x}}$ belong in two different external nodes, we take $\Delta\hat{\sigma}_{\tilde{\boldsymbol{x}}}(\boldsymbol{z}) = 0$. Instead of solving the integral in equation (9) with a direct method, we use a simple MC procedure to obtain an approximate solution. A large subset of $\boldsymbol{Z}$ is drawn from the prior distribution of the input, and $\Delta\sigma(\tilde{\boldsymbol{x}})$ is approximated by $\Delta\sigma(\tilde{\boldsymbol{x}}) = |Z|^{-1}\sum_{\boldsymbol{y}\in\boldsymbol{Z}}\Delta\hat{\sigma}^2_{\tilde{\boldsymbol{x}}}(\boldsymbol{z})$.

We sequentially sample the input $\tilde{\boldsymbol{x}}$, which has a larger $\Delta\sigma(\tilde{\boldsymbol{x}})$. Compared to ALM, ALC adaptive sampling have the advantage of not only giving more reasonable values, but it also

accounts for the distribution of the input variables. In practice, this may be very important as we can now concentrate attention on scientifically more reasonable subregions of the input space.

The above sampling technique assumes that the multiple responses are in the same scale. As pointed out by a reviewer, this sampling technique is not appropriate in cases where the outputs are expressed in different scale. To deal with this issue, we should normalize the output responses in order to be in the same scale. When we cannot normalize the output responses, other solutions should be explored in future work.

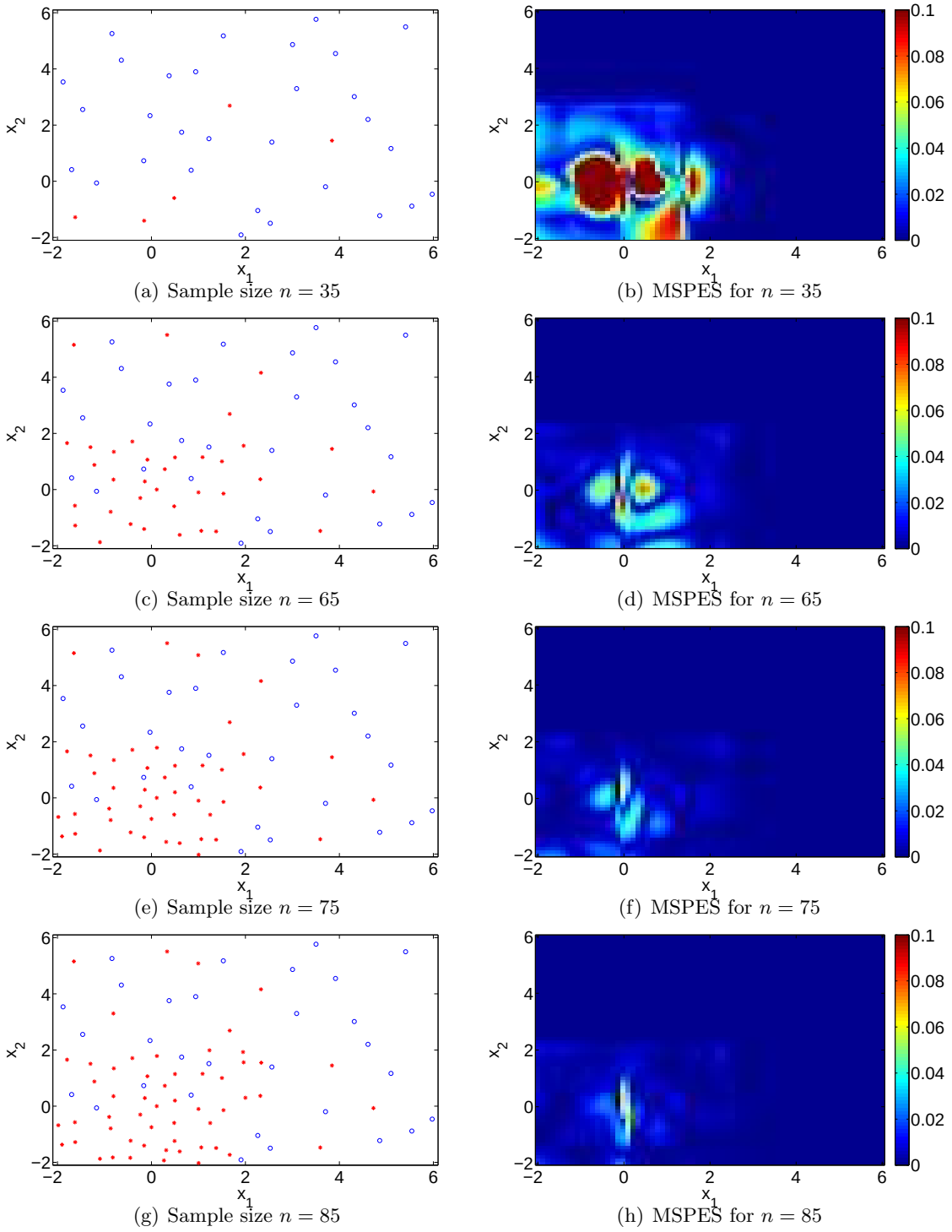
5 Artificial examples

In this section, we conduct a number of simulation studies to illustrate the performance of the proposed BTMGP and the ALC sequential adaptive sampling. We design the artificial examples so that they involve multivariate output with discontinuities and localized features. The parameters in the prior distribution of the tree are set $\alpha = 0.6$ and $\beta = 2$ as in Chipman et al. (1998). For simplicity, we use constant mean in each external leaf (subregion) of the BTMGP. After running a number of MCMC iterations, we run the ALC five times consecutively by increasing the number of points by 5 each time. We call this the ALC with BTMGP adaptive sampling technique with *Step* 5 samples. In our examples, we have not observed any significant differences between ALC with BTMGP of *Step* 5 and *Step* 1.

5.1 2-input and 2-output example

In this artificial example, we consider a two-dimensional input space $\boldsymbol{x} \in [-2, 6]^2$ and two-dimensional output functions problem $\mathbf{f}_1(\boldsymbol{x}) = x_1 \exp(-x_1^2 - x_2^2) + \epsilon_1$ and $\mathbf{f}_2(\boldsymbol{x}) = \sqrt{|x_1|} \exp(-x_1^2 - x_2^2) + \epsilon_2$, where $\epsilon_1 \equiv \epsilon_2 \sim N(0, \sigma = 0.001)$. These functions have two localized features inside the box $[-2, 2] \times [-2, 2]$, while they are practically zero everywhere else. The first function $\mathbf{f}_1(\boldsymbol{x})$ has been used in previous work by Gramacy and Lee (2009) with only one-dimensional output. The second function $\mathbf{f}_2(\boldsymbol{x})$ is chosen such that there is a dependency with function $\mathbf{f}_1(\boldsymbol{x})$, which changes from subregion to subregion. For input subregion $[-2, 0] \times [-2, 2]$, the two output functions have negative correlation, while for input subregion $[0, 2] \times [-2, 2]$, the two output functions have positive correlation. This difference should be captured from the proposed model. Moreover, the computational simplicity of these functions allows us to thoroughly test the dependence of our scheme on any number of adaptive sampling techniques.

We assume a uniform prior distribution for the input space. To ensure the posterior distribution of the parameters in each external leaf is proper, each leaf of the tree must contain at least $n_i \geq m + q + 1 = 1 + 2 + 1 = 4$ input samples.



(a) Sample size $n = 35$ (b) MSPES for $n = 35$

(c) Sample size $n = 65$ (d) MSPES for $n = 65$

(e) Sample size $n = 75$ (f) MSPES for $n = 75$

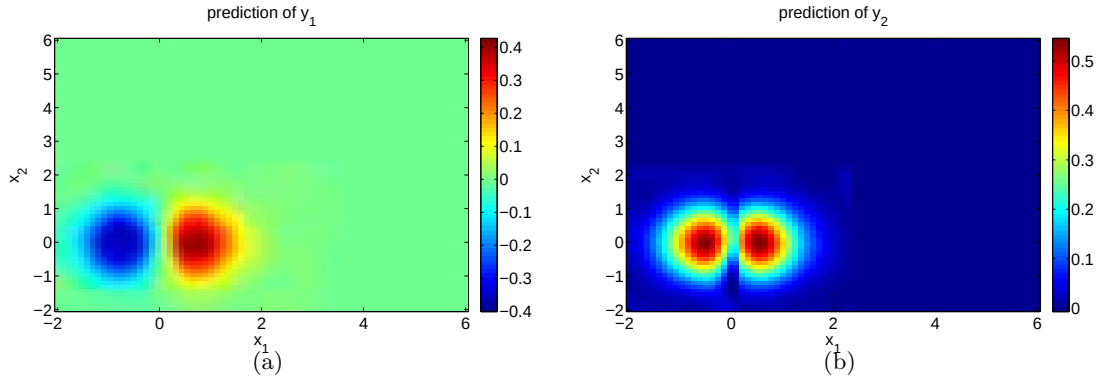
(g) Sample size $n = 85$ (h) MSPES for $n = 85$

Figure 1: Left column represents the exponential data after 35 (top), 65 (second row), 75 (third row) and 85 (bottom) adaptively chosen samples. In the right column, mean squared prediction error surface (MSPES) is plotted for the corresponding sample size in the same row.

Table 1: $\mathbf{f}_1$ and $\mathbf{f}_2$ Mean Square Prediction Error for different methods and sample sizes

		Sample size					
Method	Function	$n = 35$	$n = 45$	$n = 55$	$n = 65$	$n = 75$	$n = 85$
ALC with BTMGP	$\mathbf{f}_1$	0.0131	0.0089	0.0042	0.0026	0.0013	0.0009
	$\mathbf{f}_2$	0.0159	0.0105	0.0069	0.0043	0.0024	0.0018
ALC with BTGP	$\mathbf{f}_1$	0.0141	0.0085	0.0048	0.0024	0.0012	0.0009
	$\mathbf{f}_2$	0.0230	0.0137	0.0096	0.0074	0.0043	0.0029
LHS with BTMGP	$\mathbf{f}_1$	0.0146	0.0100	0.0092	0.0084	0.0051	0.0032
	$\mathbf{f}_2$	0.0187	0.0143	0.0129	0.0098	0.0074	0.0051
ALC with MGP	$\mathbf{f}_1$	0.0448	0.0221	0.0204	0.0098	0.0235	0.0154
	$\mathbf{f}_2$	0.1870	0.1387	0.1170	0.1364	0.1309	0.1304

We start with an initial set of 30 LHSs (blue circles), and new candidates are chosen from a sequential ALC with BTMGP adaptive sampling as previously described (red stars). The first column of Figure 1 illustrates the sequential adaptive sampling for different sample sizes. Figure 1 shows four different snapshots taken after $5, 35, 45$, and 55 ALC with BTMGP adaptive samples. The ALC with BTMGP prefers samples from the first quadrant, where variations of $\mathbf{f}_1$ and $\mathbf{f}_2$ are high, and results in higher frequency of adaptive sampling in the region, e.g., the left bottom row plot of the figure has more than 58% of the total samples located in this quadrant. The second row of the figure shows the mean square prediction error surface (MSPES), which corresponds to the sample in the first column. The MSPES is computed as the mean square error of the true value in comparison with the mean of the Bayesian predictive density function as it is described in Section 3.3, for both outputs at particular locations. As we increase the sample size, the MSPES reduces. To better visualize the response surface, we also plot the mean of the Bayesian predictive density function for sample size 85 in Figure 2. Both response surfaces demonstrate very good approximation of the predicted values with the true functions.



(a) (b)

Figure 2: Average prediction values of the Bayesian treed multivariate Gaussian process.

To demonstrate the performance of the proposed model and sampling method, we compare

it to some potential alternatives. For comparison, we use the mean square prediction error (MSPE) integrated over the whole input space. We compare the proposed ALC with BTMGP and the ALC with multiple and independent BTGP described in Gramacy and Lee (2009). This comparison illustrates the importance of modeling the dependence of the output in our multivariate analysis. We also compare the proposed method with the multivariate GP described in Conti and O'Hagan (2010) and sequentially sample the input space using the ALC as described in this paper. This comparison shows the importance of the proposed BTMGP juxtaposed with the MGP model. Moreover, to depict the usefulness of ALC with BTMGP adaptive sampling, we also compare its prediction performance to that of BTMGP when samples are chosen by LHS.

Table 1 shows the MSPE of the four different methods and two output variables $(\mathbf{f}_1, \mathbf{f}_2)$ for different sample sizes. Overall, the MSPE of ALC with BTMGP is smaller than ALC with MGP and LHS with BTMGP. The MSPE of ALC with multiple and independent BTGPs is similar to the MSPE of ALC with BTMGP for the first function $(\mathbf{f}_1)$. However, in the second function $(\mathbf{f}_2)$, the MSPE of ALC with multiple and independent BTGPs is larger than the proposed ALC with BTMGP. For example, when the sample size is 85, ALC with BTMGP produces predictions with $MSPE = 0.0009$ for $\mathbf{f}_1$ and $MSPE = 0.0017$ for $\mathbf{f}_2$. For the same sample size, the mean square error using ALC with BTGP is $MSPE = 0.0009$ for $\mathbf{f}_1$ and $MSPE = 0.0029$ for $\mathbf{f}_2$. From these differences, it is evident why the proposed method should be preferred over BTGP (Gramacy and Lee, 2008, 2009) when the outputs are correlated. For the same sample size, the mean square error using LHS with BTMGP is $MSPE = 0.0032$ for $\mathbf{f}_1$ and $MSPE = 0.0051$ for $\mathbf{f}_2$. A prediction using only a single global multivariate GP for the whole input region affords an even poorer result. Specifically, $MSPE = 0.0154$ for $\mathbf{f}_1$, and $MSPE = 0.1304$ for $\mathbf{f}_2$. The stationary assumption is violated. As such, the MGP fails to fit the data. This also is evident by the fact that even when we increase the sample size, there is not much improvement in the MSPE. More sophisticated covariance functions, such as the linear model of coregionalization (LMC) (Banerjee et al., 2004), can be used to improve predictability. However, this is beyond the scope of the present work.

One more interesting observation, shown in Table 1, is decreasing rate of MSPE as we increase the sample size. Using ALC with BTMGP has the fastest MSPE decrease rate as the sample size increases. The ALC with MGP results appear problematic as its MSPE seems to increase the sample size. Even in cases using ALC sampling, a good model should be used for the data to ensure viable results. The combination of the BTMGP model and ALC sampling achieves the best results in terms of prediction performance.

In the case of multivariate computer experiments, the proposed ALC with BTMGP model

offers an automatic and reliable way for predicting and sampling the input space within the most informative input variables. Moreover, as shown in the online supplementary material, it allows us to focus the sampling into input areas that are more interesting to domain scientists.

6 Application: regenerator of a carbon capture unit

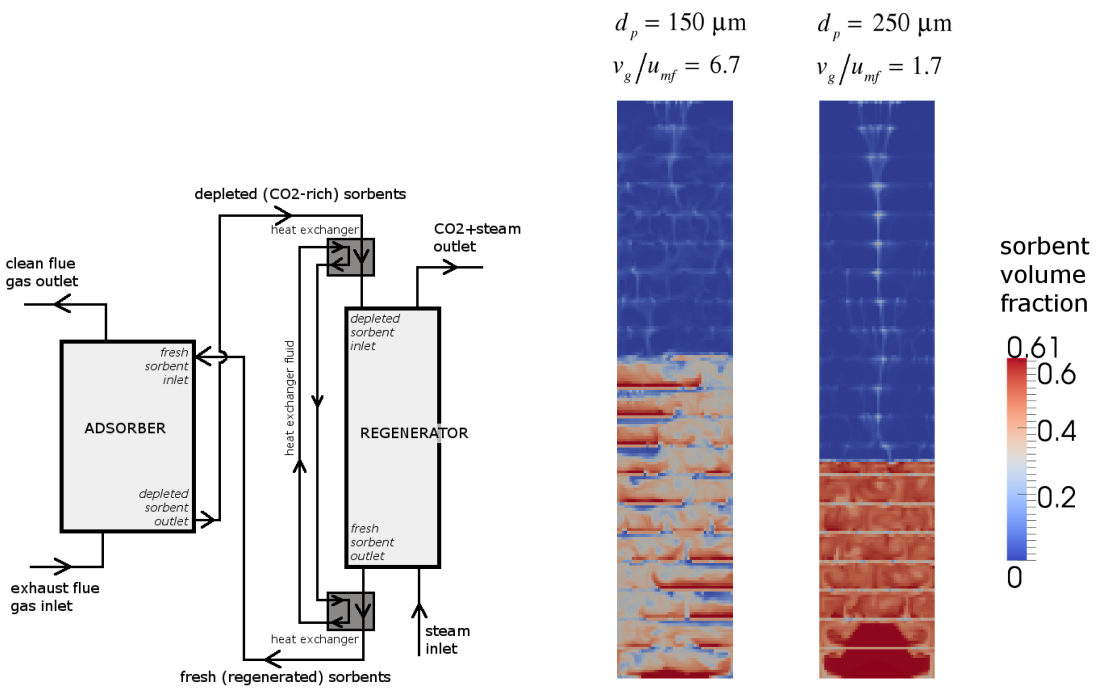
6.1 Carbon capture regenerator unit

A typical carbon capture unit consists of two devices: the adsorber and the regenerator (Figure 3(a)). Solid sorbent particles capable of reversibly reacting with carbon dioxide (CO_2) is looped through the two devices. In the adsorber, fresh sorbent particles react and trap the CO_2 from the exhaust flue gas. The depleted sorbent is then transferred to the regenerator, where the reverse chemical reaction releases the carbon dioxide back into the gaseous phase for further liquefaction and sequestration (i.e., long-term storage in deep underground reservoirs). The regenerated sorbent particles are recycled back to the adsorber.

In this study, we focus on the regenerator of the capture unit. The bulk of the energy penalty is associated with the regenerator (MacDowell et al., 2010; Gáspár and Cormoş, 2011) and therefore efforts to increase capture plant efficiency should begin with optimizing the regenerator performance. Details regarding the design and flow in the regenerator are presented in Sarkar et al. (2014). Sorbent flow inside the device is chaotic, and different regions can have varying distribution of sorbent densities (representative snapshots shown in Figure 3(b)). Variations in the local distribution of sorbents, in turn, affects carbon capture rates and device efficiency. Flow of sorbent particles in the regenerator, characterized using the density distribution of sorbent volume fraction, is sensitive to operating conditions such as the velocity of exhaust gases entering the device and the size of sorbent particles, and can sometimes change abruptly with these operating parameters (Sarkar et al., 2014). Therefore, we expect the sorbent volume fraction distribution to be non-stationary with respect to the operating conditions for the regenerator. In this work, though our BTMGP model, we will show that the output (solid fraction distribution) is indeed non-stationary with respect to the input parameters (operating conditions).

Figure 4 presents the (heterogeneous) distribution functions of the sorbent volume fraction for the two operating conditions shown in Figure 3(b). In this study, we investigate the dependence of sorbent distribution function for two input operating conditions: the particle diameter d_p (expressed in micro metres, μm), and the scaled velocity v_g/u_{mf} of gas injected at the bottom inlet (dimensionless, scaled by a characteristic velocity scale u_{mf}; see Gidaspow (1994)). At this point, with little information regarding reaction kinetics, we expect that intermediate solid volume fractions (0.2-0.4) are likely to result in better regenerator performance.

In order to determine the distribution of sorbents for a given set of operating conditions,



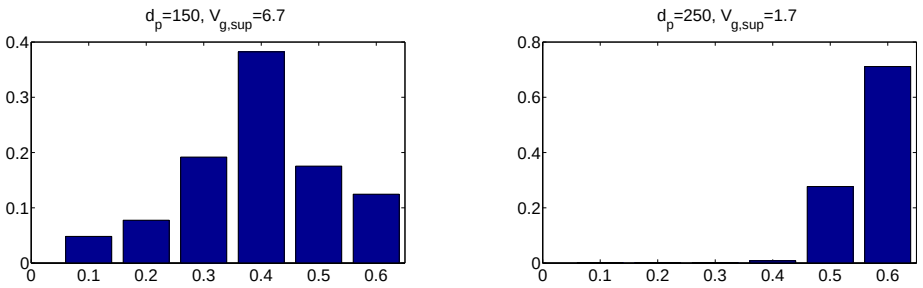
(a) Carbon Capture Unit (b) Sorbent Volume Fraction Distribution

Figure 3: (a) Schematic of a carbon capture unit with adsorber and regenerator. (b) Snapshots of the sorbent volume fraction distribution in the regenerator for two operating conditions: for different sorbent particle diameters (d_p) and scaled gas inlet velocities (v_g/u_{mf}). Blue signifies empty regions with no particles, while and red represents regions with densely packed sorbents.

computationally expensive CFD simulations may be performed. Approximately $4-7$ days of wall-time was required to complete each simulation, running in parallel on 20 processors. The high cost of these CFD models motivates us to develop a BTMGP-based surrogate model capable of predicting the solid fraction distribution in the regenerator which can be used in a sequential sampling design.

6.2 BTMGP on the regenerator of a carbon capture unit

In this section, we focus on the analysis of the sorbent distribution function in a carbon capture unit regenerator (as described in Section 6.1). We begin our analysis with the 36 available simulations of the regenerator reported in Sarkar et al. (2014). Simulations are performed for varying particle diameter d_p and scaled gas velocity v_g/u_{mf}. For our purposes, six bins are considered sufficient to characterize the distribution function of the solid volume fraction. The full solid fraction range of 0.0 to 0.6 is subdivided into six bins of fixed length, given by $[0, 0.1], (0.1, 0.2], (0.2, 0.3], (0.3, 0.4], (0.4, 0.5]$, and $(0.5, 0.6]$. As explained in Section 6.1, the frequency distribution function is a key multivariate response that affects the reaction kinetics.



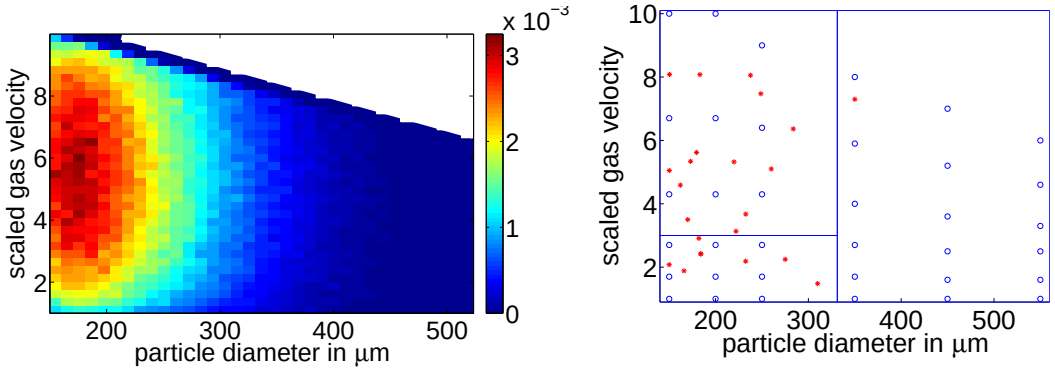
(a) $d_p = 150$ μm and $v_g/u_{mf} = 6.7$ (b) $d_p = 250$ μm and $v_g/u_{mf} = 1.7$

Figure 4: Solid fraction distribution function for the two operating cases shown in Figure 3(b).

For each of the 36 simulations, we compute the frequency distribution of the solid fraction as depicted in Figure 4. The height of the i^{th} bin represents the relative frequency $\pi_i(\boldsymbol{x})$ of the corresponding solid fraction range, where $i = 1, \ldots, 6$ and $\boldsymbol{x} = (d_p, v_g/u_{mf})$. In order to use the BTMGP model developed in Section 3, we transform the values of the relative frequency $\pi_i(\boldsymbol{x})$ from $[0, 1]$ to $(-\infty, +\infty)$. The logit (logarithm of the odds) transformation, a well-known and popular transformation with the desired properties, is used to transform $\pi_i(\boldsymbol{x})$, given by $f_i(\boldsymbol{x}) = \ln\left(\pi_i(\boldsymbol{x})/(1 - \pi_i(\boldsymbol{x}))\right)$, where i $(= 1, \ldots, 6)$ represents the i^{th} bin, $\pi_i(\boldsymbol{x})$ is the probability value of the i^{th} bin, and $\boldsymbol{x}$ represents the input variables. We can now build a BTMGP surrogate model for the vector of the logits $\mathbf{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_6(\boldsymbol{x}))^T$. Conditionally on a tree leaf, we assume that $\mathbf{f}(\,\cdot\,)$ follows a multivariate normal distributed (as in Section 2.1) equation 1 with constant mean.

Based on the knowledge gained from a preliminary analysis on the 36 initial simulations (Sarkar et al., 2014), we focus our attention on the region with particle size $d_p \in [150\,\mu\text{m}, 270\,\mu\text{m}]$ and scaled superficial gas velocity $v_g/u_{mf} \in (1.5, 9.0)$. Hence, we generate an artificial two-dimensional Beta distribution that assigns greater importance to this region of interest. Specifically, we assume two independent truncated (modified) Beta distributions in the rectangular region, given by $[150, 550] \times [1.0, 10.0]$, and exclude the upper-right corner as the operating conditions lying in that region are not of interest. Figure 5(a) depicts our assumed prior input distribution.

We begin our analysis by running a number of MCMC iterations using the 36 simulations from Sarkar et al. (2014) as the first sample. The hierarchical Bayesian model for the multivariate treed GP in the examples is defined as it is in Section 3. The parameters in the prior distribution of the tree are set $\alpha = 0.6$ and $\beta = 2$ as in Chipman et al. (1998). The mean in each external leaf (subregion) of the BTMGP is modeled as constant. After a number of MCMC iterations, we run the ALC five times while including results from 5 new simulations each time similar to Gramacy and Lee (2009) and the artificial example setting. The inputs d_p and v_g/u_{mf} for these



(a) Prior (b) 59 samples

Figure 5: (a) Prior input probability and (b) 23 ALC with BTMGP samples and the maximum a posteriori (MAP) estimation of the Bayesian tree. The existing input samples are denoted with blue circles and the sequential ALC within BTMGP samples with red stars.

5 additional simulations are chosen based on the BTMGP uncertainty coupled with the input prior distribution. Hence, in addition to the initial set of 36 simulations, 25 more simulations are performed during the five ALC runs of *Step* 5.

Along with the prior input distribution, Figure 5 shows the initial observations (blue circles), the sequential adaptive samples (red stars), and the MAP estimation of the Bayesian tree. However, 2 of the 25 additional simulations performed using Multiphase Flow with Interphase eXchanges (MFIX) failed to converge satisfactorily. Instead of 25 sequential adaptive samples, only 23 samples are included in Figure 5(b). To better understand the BTMGP in this problem, we also present the MAP estimation of the Bayesian tree calculated with 59 observations and 30000 MCMC iterations in Figure 5(b), reviling are three distinct subregions as the maximum a posteriorly (MAP) of the Bayesian tree. Scaled gas velocity between 1.5 and 4.0 appears to have 10 ALC with BTMGP samples, and it is an area where non-stationarity is observed. In contrast, for scaled gas velocity between 7.0 and 10.0, the model appears to be quite smooth, and only 4 ALC with BTMGP samples are selected.

Figure 6 shows the prediction surface of the six different probabilities π_i using BMA in a dense grid (70×70). As mentioned previously, operating conditions lying in the upper-right corner are not of interest. Therefore, no simulations are performed for those values. If good regeneration is expected for an intermediate solid fraction range of, say, 0.3 to 0.4, the areas of interest would be regions where π_4 is large. From Figure 6(d), the region where π_4 is large is given by $d_p \in (150\,\mu\text{m}, 250\,\mu\text{m})$ and scaled gas velocity $v_g/u_{mf} \in (4.0, 8.0)$. As mentioned, we pay particularly close attention to this region. Other probabilities may also be of interest. For example, we may want to avoid the regions where π_1 and π_6 are large, regions with very low or

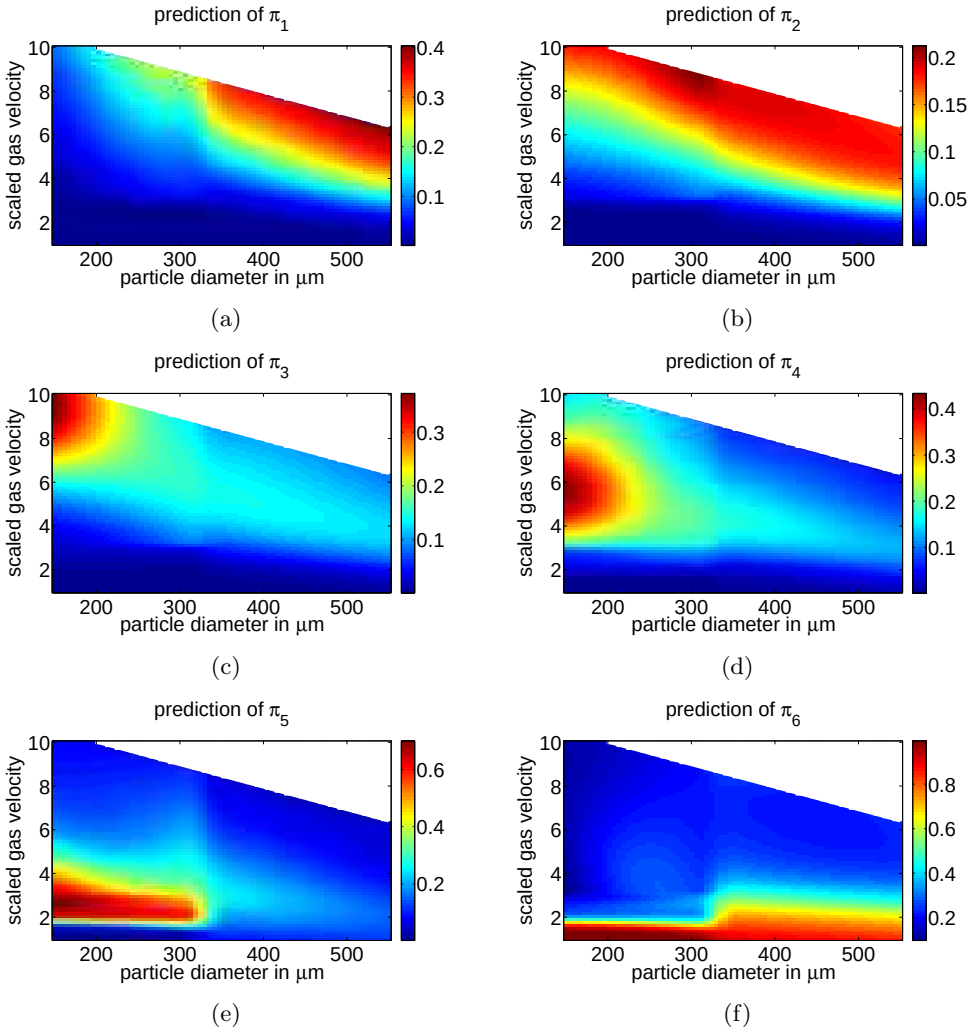


Figure 6: BMA prediction surface of the six different probabilities, as defined above, for different particle diameters in μm and scaled gas velocities.

very high solid fractions.

To better evaluate the different models' prediction abilities, a cross-validation analysis is conducted. We sample the computer code four more times for different combinations of particle diameter d_p and scaled gas velocity v_g/u_{mf}. Figure 7 shows the six bin empirical solid fraction distribution for these four different combinations. We compute and compare the predicted solid fraction distribution of four input values using the multivariate GP with separable function, the BTGP (Gramacy and Lee, 2008), and the proposed BTMGP. Figure 8 illustrate the empirical and prediction probabilities with a 95% prediction interval using the MGP with separable function (first column), the BTGP proposed by Gramacy and Lee (2008) (second column), and the proposed BTMGP (third column).

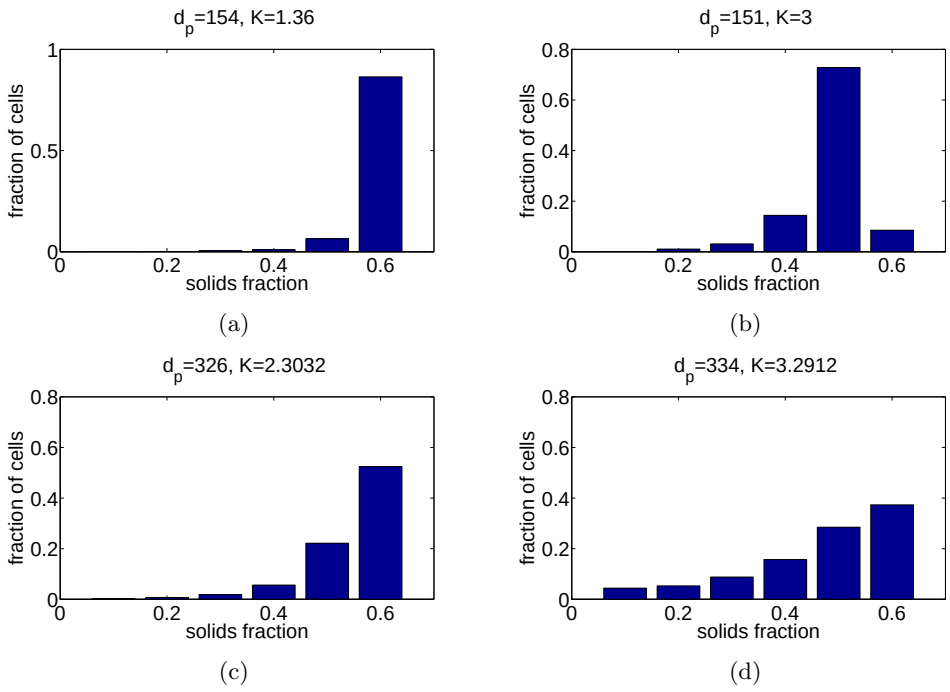


Figure 7: Real computer code results for four different combinations of particle diameter d_p and scaled gas velocity v_g/u_{mf}.

Compared with MGP, BTMGP performs better. The mean of the predictions also is closer to the real computer experiment simulation. Specifically, the MSPE using BTMGP is 0.0097, and the MSPE using MGP is 0.0216. However, predictions based on the BTMGP have larger variability due to independent assumption in the subregions associated with the Bayesian tree. Confidence intervals based on the proposed BTMGP are larger than those based on MGP. This variation can be reduced by employing Bayesian tree techniques used in Konomi et al. (2013), which defines dependent covariance functions for each subregion. However, this is beyond the scope of this paper.

In comparison with the BTGP, BTMGP performs also better. The MSPE using multiple BTGP is 0.0119, which is significantly higher than the MSPE using BTMGP. One interesting observation for the predicted solid fraction distribution with BTGP is that for the same input it has different prediction interval (PI) lengths for the six predicted bars. This is due to the fact that BTGP uses independently six simple Bayesian trees which may be different to each other. For some of the predicted bars the 95% PI lengths are similar to the BTMGP intervals. Meanwhile, for others, they are different. For example, the PI of the fifth predicted bar (π_5) of the solid fraction distribution using BTGP for input locations $(326, 2.3032)$ and $(334, 3.2912)$ have similar PI lengths to the corresponding PI lengths using BTMGP, while the sixth prediction bar (π_6) have smaller PI lengths. However, the accuracy is not necessary better. On the contrary,

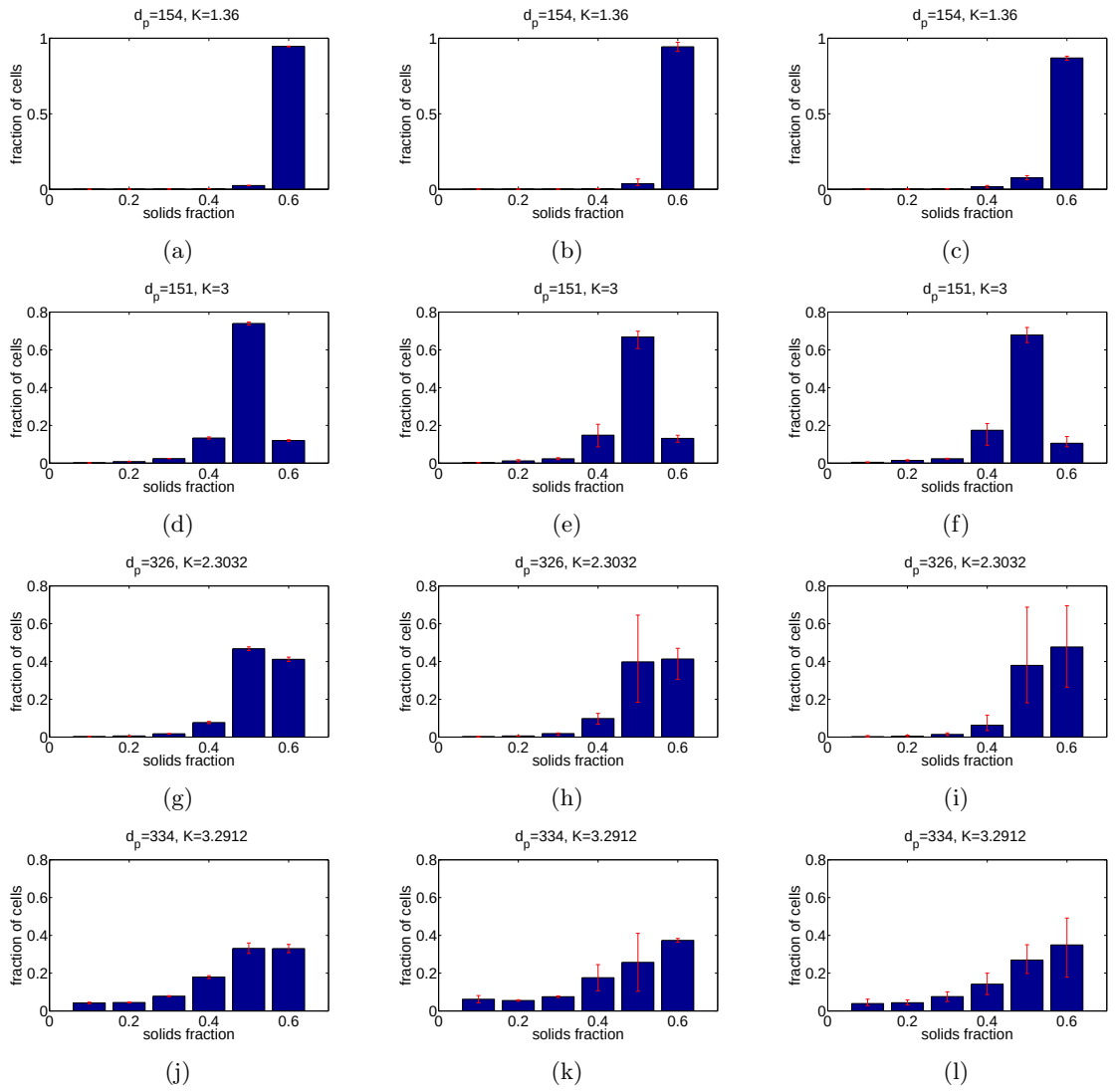


Figure 8: Prediction probabilities and their 95% prediction intervals using different models for four combinations of particle diameter d_p and scaled gas velocity v_g/u_{mf}. The first column shows the prediction probabilities and their 95% prediction intervals using multivariate GP with separable covariance function. The second column gives the prediction probabilities and their 95% prediction intervals using BTGP, and the third column provides the prediction probabilities and their 95% prediction intervals using the proposed BTMGP.

in most of the cases the mean of BTGP is misspecified, with the most remarcable difference noticed in the input $(326, 2.3032)$. The probabilities in this applicatrion are correlated, as such the BTMGP is a better model than BTGP, which assumes independent multivaraite output.

In general, we observe that prediction probabilities for input values close to available observations are more accurate than prediction for input values further away from the observations. In particular, prediction probabilities for $d_p = 151$ µm and $v_g/u_{mf} = 3$ are more accurate

than prediction probabilities for $d_p = 334$ μm and $v_g/u_{mf} = 3.2912$. We also observe that the probability predictions π_i (histogram bar heights) with values close to 0.5 tend to have larger uncertainty in comparison to π_i values close to 0 and 1. This is expected because the variance of the proportion is higher when the estimated proportion is close to 0.5.

7 Concluding remarks and extensions

Herein, we developed a Bayesian treed multivariate Gaussian process (BTMGP) that combines the Bayesian tree (Chipman et al., 1998; Gramacy and Lee, 2008) and the MGP with a separable covariance function (Mardia and Goodall, 1993; Conti and O'Hagan, 2010). The proposed BTMGP provides a multi-output emulator that can handle problems with discontinuities and localized features. The form of the separable covariance function simplifies the form of the inverse and determinant of the covariance matrix involved and, therefore, facilitates the computations within MCMC updates. Moreover, the prior specification of the MGP parameters leads to efficient local proposals in the *Grow* and *Prune* operations of the Bayesian tree. Only the parameters of the correlation function need to be updated at each MCMC iteration for prediction purposes. Moreover, we introduce numerical stability in the covariance function by adding a nugget term without increasing the computational complexity of the model.

We also propose a sequential experimental design technique based on the BTMGP predictive uncertainty. We extend the univariate ALC to properly deal with multivariate output and incorporate knowledge gained from prior studies. The proposed ALC with BTMGP adaptive sampling offers an automatic and reliable way to sample the input space and perform prediction at a low computational cost. As shown in this paper, the proposed method performs better than the multiple BTGP when the output variables are dependent.

The separable covariance function used in the proposed BTMGP can be extended into more general models such as the Linear model of Coregionalization (LMC) (Banerjee et al., 2004). Moreover, the proposed model can be extended to multiple Bayesian trees such that each output has its own Bayesian tree structure. The proposed model performs acceptably well when the output variables are dependent but one could expect a potential problematic behavior in the independent scenario. The LMC covariance function is more general than the separable covariance function and may result in better fitting. Moreover, multiple Bayesian tree may be more appropriate for some applications when some of the outputs are independent. Despite the nice features of these extensions, they involve more parameters and they are expected be computationally more expensive. A more comprehensive study of the selection of multivariate covariance functions and the use of multiple Bayesian trees is left for future research.

We applied the proposed method to the multiphase flow simulations of the full-scale regen-

erator of carbon capture system. We define the problem and transform the data to a form such that we can apply the proposed model. In our numerical example, we begin with 36 samples collected from a previous study and sequentially generate 23 more samples using ALC with BTMGP. Then, we produce predictions of the solid fraction distribution in the regenerator for different combinations of bottom inlet gas flow rate (gas velocity) and sorbent size (particle diameter). We envision that the model and computationally efficient method developed in this paper have the potential to analyze similar computer experiments. Moreover, the BTMGP can be used as a computationally efficient model to deal with high-dimension non-stationary spatial data sets with multivariate output.

Acknowledgments The research at Pacific Northwest National Laboratory (PNNL) was supported by the Department of Energy Carbon Capture Simulation Initiative. PNNL is operated by Battelle for the U.S. Department of Energy under Contract DE-AC05-76RL01830. We thank the referees and the editors for their valuable comments.

References

Banerjee, S., Carlin, B., and Gelfand, A. (2004), *Hierarchical Modeling and Analysis for Spatial Data*, Boca Raton, FL: Chapman & Hall-CRC.

Berger, J. O., Oliveira, V. D., and Sanso, B. (2001), "Objective Bayesian Analysis of Spatially Correlated Data," *Journal of the American Statistical Association*, 96, 1361–1374.

Bilionis, I., Zabaras, N., Konomi, B., and Lin, G. (2013), "Multi-output separable Gaussian process: Towards an efficient, fully Bayesian paradigm for uncertainty quantification," *Journal of Computational Physics*, 241, 212–239.

Chipman, H., George, E., and McCulloch, R. (1998), "Bayesian CART Model Search," *Journal of the American Statistical Association*, 93, 935–960.

Cohn, D. A. (1996), "Neural Network Exploration Using Optimal Experiment Design," *Advances in Neural Information Processing System*, 6, 679–686.

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

Cressie, N. (1993), *Statistics for Spatial Data. 2nd edition*, New York: John Wiley and Sons Inc.

Denison, D., Mallick, B., and Smith, A. (1998), "A Bayesian CART Algorithm," *Biometrika*, 85, 363–377.

Gáspár, J. and Cormoş, A. (2011), "Dynamic modeling and validation of absorber and desorber columns for post-combustion CO_2 capture," *Computers & Chemical Engineering*, 35, 2044–2052.

Gelfand, A. E. and Smith, A. F. M. (1990), "Sampling-Based Approaches to Calculating Marginal Densities," *Journal of the American Statistical Association*, 85, 398–409.

Gidaspow, D. (1994), *Multiphase Flow and Fluidization: Continuum and Kinetic Theory Descriptions*, Academic Press, San Diego, 1st ed.

Giles, M. B. (2008), "Multi-level Monte Carlo path simulation," *OPERATIONS RESEARCH*, 56, 607–617.

Godsill, S. (2001), "On the relationship between Markov chain Monte Carlo methods for model uncertainty," *Journal of Computational and Graphical Statistics*, 10, 230–248.

Gramacy, R. B. and Lee, H. K. H. (2008), "Bayesian treed Gaussian process Models with an application to computer modeling," *Journal of the American Statistical Association*, 103, 1119–1130.

— (2009), "Adaptive Design and Analysis of Supercomputer Experiments," *Technometrics*, 51, 130–145.

Green, P. (1995), "Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination," *Biometrika*, 82, 711–732.

Hastings, W. K. (1970), "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, 57, 97–109.

Hjort, N. and Omre, H. (1994), "Topics in Spatial Statistics [with Discussion, Comments and Rejoinder]," *Scandinavian Journal of Statistics*, 289–357.

Iman, R. L. and Conover, W. J. (1980), "Small sample sensitivity analysis techniques for computer models, with an application to risk assessment," *Communications in Statistics - Theory and Methods*, 9.

Karagiannis, G. and Andrieu, C. (2013), "Annealed Importance Sampling Reversible Jump MCMC Algorithms," *Journal of Computational and Graphical Statistics*, 22, 623–648.

Kim, H.-M., Mallick, B., and Holmes, C. (2005), "Analyzing nonstationary spatial data using a piecewise Gaussian process," *Journal of the American Statistical Association*, 470, 653–658.

Konomi, B., Sang, H., and Mallick, B. (2013), "Adaptive Bayesian nonstationary modeling for large spatial datasets using covariance approximations," *Journal of Computational and Graphical Statistics, doi: 10.1080/10618600.2013.812872*.

Liu, J., Wong, W., and Kong, A. (1994), "Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes," *Biometrika*, 81, 27–40.

MacDowell, N., Florin, N., Buchard, A., Hallett, J., Galindo, A., Jackson, G., Adjiman, C., Williams, C., Shah, N., and Fennell, P. (2010), "An overview of CO_2 capture technologies," *Energy & Environmental Science*, 3, 1645–1669.

Mardia, K. V. and Goodall, C. R. (1993), "Spatial temporal analysis of multivariate environmental monitoring data," *In Multivariate Environmental Statistics (G. P. Patil and C. R. Rao, eds.)*, 347–386.

Mueller, P. (1993), "Alternatives to the Gibbs sampling scheme," Tech. rep., Institute Statistics and Decision Sciences, Duke University.

Sarkar, A., Pan, W., Suh, D., Huckaby, E., and Sun, X. (2014), "Multiphase flow simulations of a moving fluidized bed regenerator in a carbon capture unit," *Powder Technology*, In press.

Seo, S., Wallat, M., Graepel, T., and Obermayer, K. (2000), "Gaussian Process Regression: Active Data Selection and Test Point Rejection." *IEEE. In Proceedings of the International Joint Conference on Neural Networks*, III, 241–246.