

Scalable Variational Bayesian Kernel Selection for Sparse Gaussian Process Regression

Tong Teng,^{1*} Jie Chen,^{2*} Yehong Zhang,^{1*} Bryan Kian Hsiang Low¹

¹Department of Computer Science, National University of Singapore, Republic of Singapore

²College of Computer Science and Software Engineering, Shenzhen University, P. R. China
{tengtong, yehong, lowkh}@comp.nus.edu.sg, chenjie@szu.edu.cn

Abstract

This paper presents a *variational Bayesian kernel selection* (VBKS) algorithm for *sparse Gaussian process regression* (SGPR) models. In contrast to existing GP kernel selection algorithms that aim to select only one kernel with the highest model evidence, our VBKS algorithm considers the kernel as a random variable and learns its belief from data such that the uncertainty of the kernel can be interpreted and exploited to avoid overconfident GP predictions. To achieve this, we represent the probabilistic kernel as an additional variational variable in a *variational inference* (VI) framework for SGPR models where its posterior belief is learned together with that of the other variational variables (i.e., inducing variables and kernel hyperparameters). In particular, we transform the discrete kernel belief into a continuous parametric distribution via reparameterization in order to apply VI. Though it is computationally challenging to jointly optimize a large number of hyperparameters due to many kernels being evaluated simultaneously by our VBKS algorithm, we show that the variational lower bound of the log-marginal likelihood can be decomposed into an additive form such that each additive term depends only on a disjoint subset of the variational variables and can thus be optimized independently. Stochastic optimization is then used to maximize the variational lower bound by iteratively improving the variational approximation of the exact posterior belief via stochastic gradient ascent, which incurs constant time per iteration and hence scales to big data. We empirically evaluate the performance of our VBKS algorithm on synthetic and massive real-world datasets.

1 Introduction

A *Gaussian process regression* (GPR) model is a kernel-based Bayesian nonparametric model that represents the correlation/similarity of the data using a kernel for performing nonlinear probabilistic regression. A limitation of such a *full-rank* GPR model is its poor scalability to big data since computing its predictive belief and learning the kernel hyperparameters incur cubic time in the data size. To overcome this limitation, a number of *sparse GPR* (SGPR) models have been proposed (Chen et al. 2012; 2013; 2015; Gal

and Turner 2015; Hensman et al. 2015; Hoang, Hoang, and Low 2015; 2016; 2017; Lázaro-Gredilla et al. 2010; Low et al. 2015a; 2015b; Quiñero-Candela and Rasmussen 2005; Titsias 2009; Titsias and Lázaro-Gredilla 2013; Xu et al. 2014; Yu et al. 2019b). These SGPR models exploit a small set of inducing variables or a spectral representation of the kernel to derive a low-rank GP approximation for achieving scalable training and prediction to big data.

All the SGPR models mentioned above are either designed only for the widely-used *squared exponential* (SE) kernel (Titsias and Lázaro-Gredilla 2013; Yu et al. 2019b) or assume the kernel type to be specified by the user *a priori*. However, in the era of big data, it has become all but impossible for non-experts to manually select an appropriate kernel for a GP model since the underlying correlation structures of massive datasets are usually too complex to be captured by the commonly-used base kernels (e.g., SE and periodic kernels). Such an issue can be resolved by kernel selection algorithms (Duvenaud et al. 2013; Kim and Teh 2018; Lu et al. 2018; Malkomes, Schaff, and Garnett 2016) which are designed to automatically find a kernel with the highest model evidence (e.g., marginal likelihood, Bayesian information criterion) given a dataset. These algorithms have demonstrated success in improving the GP predictive performance over that of using the manually specified kernel. But, selecting only one kernel with the highest model evidence and ignoring the uncertainty of it being the true kernel may result in overconfident predictions, especially if other kernels yield similar model evidences (Hoeting et al. 1999). This motivates the need to design and develop a *Bayesian kernel selection* (BKS) algorithm that, instead of searching for the *best* kernel, considers the kernel as a random variable defined over a kernel space and learns a belief of the probabilistic kernel from data such that the uncertainty of the kernel can be interpreted and exploited to avoid overconfident GP predictions, which is the focus of our work here.

Most existing BKS algorithms for GP models approximate the kernels using their spectral density representation and thus only work for stationary kernels (Benton et al. 2019; Oliva et al. 2016; Wilson and Adams 2013). The BKS algorithm of Malkomes and Garnett (2015) caters to any kernel but is designed for the *full-rank* GPR model

*Equal contribution.

only. So, its approach of updating the kernel belief scales poorly in the data size. This paper presents a *variational BKS* (VBKS) algorithm for SGPR models without any stationary assumption on the kernels. In particular, we represent the probabilistic kernel as an additional variational variable in a *variational inference* (VI) framework for the SGPR models where its posterior belief is learned together with that of the other variational variables (i.e., inducing variables and hyperparameters) by minimizing the *Kullback-Leibler* (KL) divergence between a variational approximation and their exact posterior belief or, equivalently, maximizing a variational lower bound of the log-marginal likelihood (Section 3). Unfortunately, the existing variational SGPR models (Hensman et al. 2015; Hoang, Hoang, and Low 2015; 2016; Titsias 2009; Titsias and Lázaro-Gredilla 2013; Yu et al. 2019b) cannot be straightforwardly used by our VBKS algorithm since (a) they commonly use continuous distributions (e.g., normal distribution) that cannot directly accommodate the discrete kernel belief involving further constraints, and (b) the VBKS algorithm has to jointly learn the posterior belief of a large number of hyperparameters from big data as *many* kernels of *different* types are evaluated simultaneously, which is computationally more expensive than approximating the posterior belief of hyperparameters for only one specified kernel.

To address the above challenges, we first reparameterize the discrete kernel belief using variational variables to a continuous parametric distribution (Section 3) and then decompose the variational lower bound into an additive form such that each additive term depends only on a disjoint subset of the variational variables and can thus be maximized independently (Section 4). To maximize the variational lower bound, stochastic optimization is used to iteratively improve the variational approximation of the exact posterior belief via stochastic gradient ascent where the stochastic gradient is estimated by first sampling from the variational distributions and then from the observed data (Section 4.1). The former sampling step makes the gradient estimation applicable to any differentiable kernel function in the kernel space (rather than only the SE kernel adopted in the works of Titsias and Lázaro-Gredilla (2013) and Yu et al. (2019b)) while the latter step enables our VBKS algorithm to incur constant time per iteration and hence scale to big data. Consequently, the kernel posterior belief can be updated as and when more training data is used, which makes it possible to perform BKS without using the full massive dataset and thus achieve competitive predictive performance fast. We empirically evaluate the performance of our VBKS algorithm on synthetic and two massive real-world datasets.

2 Background and Notations

2.1 Gaussian Process Regression (GPR)

Let \mathcal{X} denote a d -dimensional input domain such that each input vector $\mathbf{x} \in \mathcal{X}$ is associated with a noisy output $y(\mathbf{x}) \triangleq f(\mathbf{x}) + \epsilon$ observed from corrupting the function f evaluated at \mathbf{x} by an additive noise $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ where σ_n^2 is the noise variance. Let $\{f(\mathbf{x})\}_{\mathbf{x} \in \mathcal{X}}$ denote a *Gaussian process* (GP), that is, every finite subset of $\{f(\mathbf{x})\}_{\mathbf{x} \in \mathcal{X}}$

follows a multivariate Gaussian distribution. Such a GP is fully specified by its *prior* mean $\mathbb{E}[f(\mathbf{x})]$ and covariance $k(\mathbf{x}, \mathbf{x}') \triangleq \text{cov}[f(\mathbf{x}), f(\mathbf{x}')] for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, the latter of which is usually defined by one of the widely-used kernels (e.g., *squared exponential* (SE), *periodic* (PER)) with a vector of hyperparameters $\boldsymbol{\theta}_k$. In this paper, $\mathbb{E}[f(\mathbf{x})]$ is assumed to be zero and $f(\mathbf{x}; k)$ is used to denote a function $f(\mathbf{x})$ with GP prior covariance $k(\mathbf{x}, \mathbf{x}')$ for notational simplicity.$

Supposing a column vector $\mathbf{y}_{\mathcal{D}} \triangleq (y(\mathbf{x}))_{\mathbf{x} \in \mathcal{D}}^{\top}$ of noisy outputs are observed by evaluating function f at a set $\mathcal{D} \subset \mathcal{X}$ of training inputs, a full-rank GPR model can perform probabilistic regression by providing a GP predictive belief $p(f(\mathbf{x}_*)|\mathbf{y}_{\mathcal{D}}) \triangleq \mathcal{N}(\mu_{\mathbf{x}_*|\mathcal{D}}, \sigma_{\mathbf{x}_*|\mathcal{D}}^2)$ for any test input $\mathbf{x}_* \in \mathcal{X}$ with the following *posterior* mean and variance:

$$\begin{aligned} \mu_{\mathbf{x}_*|\mathcal{D}} &\triangleq \Sigma_{\mathbf{x}_*\mathcal{D}}(\Sigma_{\mathcal{D}\mathcal{D}} + \sigma_n^2 I)^{-1} \mathbf{y}_{\mathcal{D}} \\ \sigma_{\mathbf{x}_*|\mathcal{D}}^2 &\triangleq k(\mathbf{x}_*, \mathbf{x}_*) - \Sigma_{\mathbf{x}_*\mathcal{D}}(\Sigma_{\mathcal{D}\mathcal{D}} + \sigma_n^2 I)^{-1} \Sigma_{\mathcal{D}\mathbf{x}_*} \end{aligned} \quad (1)$$

where $\Sigma_{\mathbf{x}_*\mathcal{D}} \triangleq (k(\mathbf{x}_*, \mathbf{x}))_{\mathbf{x} \in \mathcal{D}}$, $\Sigma_{\mathcal{D}\mathcal{D}} \triangleq (k(\mathbf{x}, \mathbf{x}'))_{\mathbf{x}, \mathbf{x}' \in \mathcal{D}}$, and $\Sigma_{\mathcal{D}\mathbf{x}_*} \triangleq \Sigma_{\mathbf{x}_*\mathcal{D}}^{\top}$. Due to the inversion of $\Sigma_{\mathcal{D}\mathcal{D}} + \sigma_n^2 I$, computing the above predictive belief incurs $\mathcal{O}(|\mathcal{D}|^3)$ time and thus scales poorly in the size $|\mathcal{D}|$ of observed data.

2.2 Sparse Gaussian Process Regression (SGPR)

To improve the scalability of the GP model, a number of SGPR models have been proposed. These SGPR models exploit a vector $\mathbf{u}_k \triangleq \{f(\mathbf{x}; k)\}_{\mathbf{x} \in \mathcal{U}}$ of inducing variables¹ for a small set $\mathcal{U} \subset \mathcal{X}$ of inducing inputs (i.e., $|\mathcal{U}| \ll |\mathcal{D}|$) for approximating the GP predictive belief:

$$\begin{aligned} p(f(\mathbf{x}_*)|\mathbf{y}_{\mathcal{D}}) &= \int p(f(\mathbf{x}_*)|\boldsymbol{\alpha}, \mathbf{y}_{\mathcal{D}}) p(\boldsymbol{\alpha}|\mathbf{y}_{\mathcal{D}}) d\boldsymbol{\alpha} \\ &\approx \int q(f(\mathbf{x}_*)|\boldsymbol{\alpha}, \mathbf{y}_{\mathcal{D}}) q(\boldsymbol{\alpha}) d\boldsymbol{\alpha} \end{aligned} \quad (2)$$

where $\boldsymbol{\alpha}$ is a vector of variables that can be set as either $\boldsymbol{\alpha} \triangleq \mathbf{u}_k$ (i.e., $\boldsymbol{\theta}_k$ is assumed to be a point estimate) (Hoang, Hoang, and Low 2015; 2016; Quiñonero-Candela and Rasmussen 2005; Titsias 2009) or $\boldsymbol{\alpha} \triangleq \text{vec}(\mathbf{u}_k, \boldsymbol{\theta}_k)$ (Hensman et al. 2015; Titsias and Lázaro-Gredilla 2013; Yu et al. 2019b). Variational inference has been used to derive $q(\boldsymbol{\alpha})$ by minimizing the KL divergence between $q(\boldsymbol{\alpha})$ and the exact posterior belief $p(\boldsymbol{\alpha}|\mathbf{y}_{\mathcal{D}})$. Various conditional independence assumptions of $f(\mathbf{x}_*)$ and $\mathbf{y}_{\mathcal{D}}$ given $\boldsymbol{\alpha}$ have been imposed for computing $q(f(\mathbf{x}_*)|\boldsymbol{\alpha}, \mathbf{y}_{\mathcal{D}})$, which result in different sparse GP approximations (Hoang, Hoang, and Low 2015; 2016; Quiñonero-Candela and Rasmussen 2005).

2.3 Automatic Kernel Selection

All the GPR and SGPR models mentioned above assume the kernel type $k(\mathbf{x}, \mathbf{x}')$ to be specified by the user and learn \mathbf{u}_k and $\boldsymbol{\theta}_k$ from the data. However, the kernel choice is critical to the performance of the (sparse) GP models since various kernel types can capture different underlying correlation structures of the data (see Chapter 4 in (Rasmussen and Williams 2006) for a detailed discussion of various kernels).

¹Let \mathbf{u}_k denote a vector of inducing variables whose prior covariance is computed using the kernel function $k(\mathbf{x}, \mathbf{x}')$.

Let \mathcal{K} be a set of candidate kernels (e.g., SE, PER). The automatic kernel selection algorithms (Duvenaud et al. 2013; Kim and Teh 2018; Lloyd et al. 2014; Lu et al. 2018; Malkomes, Schaff, and Garnett 2016) aim to automatically find a kernel $k \in \mathcal{K}$ with the highest model evidence (e.g., marginal likelihood, Bayesian information criterion). Since the sum or product of two valid kernels (i.e., positive semidefinite kernels that define valid covariance functions) is still a valid covariance function, the kernel space \mathcal{K} can be constructed by repeatedly applying the following composition rules:

$$\begin{aligned} k_3(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}') \\ k_4(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}') \times k_2(\mathbf{x}, \mathbf{x}') \end{aligned}$$

where k_1 and k_2 can be either one of the base kernels (i.e., SE, PER, *linear* (LIN), and *rational-quadratic* (RQ)) or a composite kernel (Duvenaud et al. 2013).

3 Variational Bayesian Kernel Selection (VBKS) for SGPR Models

As mentioned in Section 1, most existing kernel selection algorithms aim to find only *one* kernel $k \in \mathcal{K}$ with the highest model evidence (Duvenaud et al. 2013; Kim and Teh 2018; Lloyd et al. 2014; Lu et al. 2018; Malkomes, Schaff, and Garnett 2016). However, if several kernels achieve similar model evidences, ignoring the uncertainty among the kernels and selecting only one kernel with the highest model evidence may result in overconfident inferences/predictions or overfitting, especially if some composite kernel structures in the kernel space are complex. To resolve this issue, the *Bayesian kernel selection* (BKS) problem considers k as a random variable and introduces a kernel belief $p(k)$ over k . Then, the GP predictive belief (2) has to consider an additional variable k of the kernel, which yields

$$p(f(\mathbf{x}_*)|\mathcal{Y}_{\mathcal{D}}) = \mathbb{E}_{p(\mathbf{f}_{\mathcal{D}}, \alpha, k|\mathcal{Y}_{\mathcal{D}})} [p(f(\mathbf{x}_*)|\mathbf{f}_{\mathcal{D}}, \alpha, k)] \quad (3)$$

where $\alpha \triangleq \text{vec}(\mathbf{u}_k, \boldsymbol{\theta}_k)$, $\mathbf{f}_{\mathcal{D}} \triangleq (f(\mathbf{x}))_{\mathbf{x} \in \mathcal{D}}^{\top}$, and \mathbf{u}_k and $\boldsymbol{\theta}_k$ still, respectively, denote the vectors of inducing variables and hyperparameters of kernel k to ease notations, even though k is now probabilistic. The exact definitions of \mathbf{u}_k , $\boldsymbol{\theta}_k$, and $p(k)$ will be introduced later. Next, the key issue is to approximate the intractable posterior belief $p(\mathbf{f}_{\mathcal{D}}, \alpha, k|\mathcal{Y}_{\mathcal{D}})$ in (3) such that the GP predictive belief $p(f(\mathbf{x}_*)|\mathcal{Y}_{\mathcal{D}})$ can be computed analytically and efficiently. To achieve this, the active structure discovery algorithm of Malkomes and Garnett (2015) has proposed to approximate $p(\boldsymbol{\theta}_k|\mathcal{Y}_{\mathcal{D}}, k)$ and $p(\mathcal{Y}_{\mathcal{D}}|k)$ via Laplace approximation such that the posterior belief of the kernel $p(k|\mathcal{Y}_{\mathcal{D}})$ can be computed by applying Bayes' rule. However, such a BKS algorithm is designed for the full-rank GPR model only. So, it still incurs $\mathcal{O}(|\mathcal{D}|^3)$ time and scales poorly in the size $|\mathcal{D}|$ of observed data.

To scale BKS of GP models to big data, we propose to approximate the posterior belief $p(\mathbf{f}_{\mathcal{D}}, \alpha, k|\mathcal{Y}_{\mathcal{D}})$ in (3) via *variational inference* (VI) for SGPR models, which we call *variational BKS* (VBKS)². In particular, the intractable pos-

terior belief $p(\mathbf{f}_{\mathcal{D}}, \alpha, k|\mathcal{Y}_{\mathcal{D}})$ in (3) can be approximated with a variational distribution:

$$q(\mathbf{f}_{\mathcal{D}}, \alpha, k) \triangleq p(\mathbf{f}_{\mathcal{D}}|\alpha, k) q(\alpha) q(k) \quad (4)$$

by minimizing the KL divergence between $q(\mathbf{f}_{\mathcal{D}}, \alpha, k)$ and the exact posterior belief $p(\mathbf{f}_{\mathcal{D}}, \alpha, k|\mathcal{Y}_{\mathcal{D}})$. The variational approximation in (4) is similar to those used by existing VI frameworks of SGPR models (Titsias and Lázaro-Gredilla 2013; Yu et al. 2019b) except that an additional term $q(k)$ is included due to the probabilistic kernel k whose posterior belief needs to be learned from data together with that of α . We will consider a finite kernel space \mathcal{K} and hence a discrete distribution over k in our work here.

Let $\mathcal{K} \triangleq \{k_i\}_{i=1}^K$ be a set of kernels where each k_i represents a kernel which can either be the base kernel or the composite kernel, as described in Section 2.3. The kernel belief $p(k)$ can be defined as a vector $\mathbf{p} \triangleq (p_i)_{i=1}^K$ where $p_i \triangleq p(k = k_i)$ for $i = 1, \dots, K$. Similarly, the variational distribution $q(k)$ can be represented by a vector $\mathbf{q} \triangleq (q_i)_{i=1}^K$ of variational parameters where $q_i \triangleq q(k = k_i)$ for $i = 1, \dots, K$. Then, the objective of VBKS is to minimize $\text{KL}[q(\mathbf{f}_{\mathcal{D}}, \alpha, k)||p(\mathbf{f}_{\mathcal{D}}, \alpha, k|\mathcal{Y}_{\mathcal{D}})]$ w.r.t. \mathbf{q} and the variational parameters of $q(\alpha)$ with the following constraints:

$$0 \leq q_i \leq 1 \text{ for } i = 1, \dots, K, \text{ and } \sum_{i=1}^K q_i = 1.$$

A commonly-used method to solve such a constrained optimization problem is to convert it to that of unconstrained optimization via some tricks (e.g., substitution, Lagrange multiplier). In this work, we achieve this by introducing a K -dimensional vector of continuous variables $\mathbf{g} \in \mathbb{R}^K$ and reparameterizing p_i using

$$p(k_i|\mathbf{g}) \triangleq \exp(g_i) / \sum_{j=1}^K \exp(g_j)$$

where g_i is the i -th element of \mathbf{g} . Then, let the variational distribution $q(k) \triangleq p(k|\mathbf{g}) q(\mathbf{g})$. The above-mentioned constrained optimization problem is transformed to that of minimizing $\text{KL}[q(\mathbf{f}_{\mathcal{D}}, \alpha, k, \mathbf{g})||p(\mathbf{f}_{\mathcal{D}}, \alpha, k, \mathbf{g}|\mathcal{Y}_{\mathcal{D}})]$ w.r.t. the variational parameters (i.e., detailed later) of $q(\alpha)$ and $q(\mathbf{g})$ without any constraints. An additional benefit of introducing \mathbf{g} as a vector of variational variables is that we can then place parametric multivariate distributions on $p(\mathbf{g})$ and $q(\mathbf{g})$ such that the prior knowledge of the kernel set can be included in $p(\mathbf{g})$ and the true correlations between different kernels can be learned from data by learning the covariance parameters of $q(\mathbf{g})$, which is useful in interpreting the relationship between kernels (e.g., a high correlation between g_i and g_j can be interpreted as a high similarity between k_i and k_j with potentially similar learning performances).

Then, minimizing $\text{KL}[q(\mathbf{f}_{\mathcal{D}}, \alpha, k, \mathbf{g})||p(\mathbf{f}_{\mathcal{D}}, \alpha, k, \mathbf{g}|\mathcal{Y}_{\mathcal{D}})]$ is equivalent to maximizing a variational *evidence lower bound* (ELBO):

$$\begin{aligned} \mathcal{L}(q) \triangleq & \mathbb{E}_{q(\mathbf{f}_{\mathcal{D}}, \alpha, k, \mathbf{g})} [\log p(\mathcal{Y}_{\mathcal{D}} | \mathbf{f}_{\mathcal{D}})] \\ & - \text{KL}[q(\mathbf{f}_{\mathcal{D}}, \alpha, k, \mathbf{g})||p(\mathbf{f}_{\mathcal{D}}, \alpha, k, \mathbf{g})] \end{aligned} \quad (5)$$

since the log marginal likelihood $\log p(\mathcal{Y}_{\mathcal{D}}) = \mathcal{L}(q) + \text{KL}[q(\mathbf{f}_{\mathcal{D}}, \alpha, k, \mathbf{g})||p(\mathbf{f}_{\mathcal{D}}, \alpha, k, \mathbf{g}|\mathcal{Y}_{\mathcal{D}})]$ is a constant. The derivation of (5) is in Appendix A. Unfortunately, the evaluation of $\mathcal{L}(q)$ is intractable since it contains the inverse

²Though our proposed VBKS algorithm performs Bayesian kernel inference instead of “selecting” a specific kernel(s), we call it “kernel selection” to be consistent with the *Bayesian model selection* framework (Rasmussen and Williams 2006).

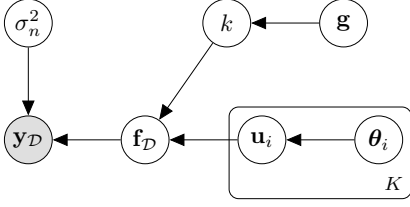


Figure 1: Graphical model of our SGPR model with the probabilistic kernel.

of the prior covariance matrix of the inducing variables \mathbf{u}_k which depends on θ_k but cannot be analytically integrated over θ_k . Some works (Titsias and Lázaro-Gredilla 2013; Yu et al. 2019b) have resolved this issue by introducing a standardized kernel and reparameterizing the prior covariance matrix such that its inversion does not depend on the hyperparameters. However, such a reparameterization trick cannot be applied to our work here since the standardized kernel is only defined for the SE kernel and cannot be generalized to cater to the other kernels, especially the composite ones. The doubly stochastic VI framework of Titsias and Lázaro-Gredilla (2014) has proposed to generalize the optimization of $\mathcal{L}(q)$ to any differentiable kernel function by sampling from the variational distribution, which cannot be straightforwardly used to optimize $\mathcal{L}(q)$ in VBKS since it is designed for GP models with only one kernel and does not consider the scalability in K when *many* kernels have to be evaluated simultaneously. Next, a scalable stochastic optimization method for VBKS will be introduced to circumvent the above-mentioned issues.

4 Scalable Stochastic Optimization for VBKS

Let $\theta_k \triangleq (\theta_i)_{i=1}^K$ and $\mathbf{u}_k \triangleq (\mathbf{u}_i)_{i=1}^K$ where θ_i are the hyperparameters of kernel k_i and $\mathbf{u}_i \triangleq (f(\mathbf{x}; k_i))_{\mathbf{x} \in \mathcal{U}}$ is a vector of inducing variables whose prior covariance is computed using kernel k_i . Optimizing $\mathcal{L}(q)$ jointly w.r.t. θ_k and \mathbf{u}_k is computationally challenging since they are both high-dimensional, especially if K is large. To improve the scalability of the optimization in K , we assume that (a) $\alpha_i \triangleq \text{vec}(\mathbf{u}_i, \theta_i)$ for $i = 1, \dots, K$ are independent and also independent of k and \mathbf{g} , and (b) \mathbf{f}_D is conditionally independent of \mathbf{g} given k . Then,

$$p(\mathbf{f}_D, \alpha, k, \mathbf{g}) = p(\mathbf{f}_D | \alpha, k) p(k | \mathbf{g}) p(\mathbf{g}) \prod_{i=1}^K p(\alpha_i). \quad (6)$$

The graphical model in Fig. 1 shows the relationship between the variables of our SGPR model with the probabilistic kernel. Let the variational distribution be factorized as

$$q(\mathbf{f}_D, \alpha, k, \mathbf{g}) = p(\mathbf{f}_D | \alpha, k) p(k | \mathbf{g}) q(\mathbf{g}) \prod_{i=1}^K q(\alpha_i) \quad (7)$$

where $p(\mathbf{f}_D | \alpha, k)$ and $p(k | \mathbf{g})$ are the exact posterior beliefs of \mathbf{f}_D and k . The ELBO $\mathcal{L}(q)$ in (5) can be decomposed as

$$\mathcal{L}(q) = \mathbb{E}_{q(\mathbf{g})} \left[\sum_{i=1}^K p(k_i | \mathbf{g}) \mathcal{L}_i(q) \right] - \text{KL}[q(\mathbf{g}) || p(\mathbf{g})] \quad (8)$$

where

$$\mathcal{L}_i(q) \triangleq \mathbb{E}_{p(\mathbf{f}_D | \alpha_i, k_i) q(\alpha_i)} [\log p(\mathbf{y}_D | \mathbf{f}_D)] - \text{KL}[q(\alpha_i) || p(\alpha_i)] \quad (9)$$

includes only the variational variables α_i and is thus called a *local ELBO*. The derivation of (8) is in Appendix B. Interestingly, to maximize (8), we can maximize each local ELBO $\mathcal{L}_i(q)$ over $q(\alpha_i)$ independently for $i = 1, \dots, K$ and then maximize (8) over $q(\mathbf{g})$ since $p(k_i | \mathbf{g}) \geq 0$ and $\mathcal{L}_i(q)$ is independent of \mathbf{g} and the variational variables of $\mathcal{L}_j(q)$ for $j \neq i$, which makes the optimization of $\mathcal{L}(q)$ incur linear time in the kernel size K and thus easily parallelizable. Next, we will discuss how to maximize $\mathcal{L}(q)$ and each $\mathcal{L}_i(q)$ via stochastic gradient ascent, which incurs only constant time per iteration and hence makes our VBKS algorithm scale well to big data.

4.1 Stochastic Optimization

In this section, we will first discuss the optimization of the local ELBOs $\mathcal{L}_i(q)$ for $i = 1, \dots, K$ and then the optimization of $\mathcal{L}(q)$ via *stochastic gradient ascent* (SGA).

Optimizing the local ELBOs. To optimize $\mathcal{L}_i(q)$ via stochastic optimization, we reparameterize the variational variables \mathbf{u}_i and θ_i by assuming that they are affine transformations of a vector of variables which follow a standard distribution. In particular, let $\mathbf{u}_i \triangleq \mathbf{C}_{\mathbf{u}_i} \boldsymbol{\eta}_{\mathbf{u}_i} + \mathbf{m}_{\mathbf{u}_i}$ and $\theta_i \triangleq \mathbf{C}_{\theta_i} \boldsymbol{\eta}_{\theta_i} + \mathbf{m}_{\theta_i}$ where $\Phi_i \triangleq \text{vec}(\mathbf{C}_{\mathbf{u}_i}, \mathbf{m}_{\mathbf{u}_i}, \mathbf{C}_{\theta_i}, \mathbf{m}_{\theta_i})$ are variational parameters that are independent of \mathbf{u}_i and θ_i , and $\boldsymbol{\eta}_i \triangleq \text{vec}(\boldsymbol{\eta}_{\mathbf{u}_i}, \boldsymbol{\eta}_{\theta_i})$ follows a standard multivariate Gaussian distribution³: $q(\boldsymbol{\eta}_i) \approx p(\boldsymbol{\eta}_i | \mathbf{y}_D) \triangleq \mathcal{N}(\boldsymbol{\eta}_i | \mathbf{0}, I)$. We can factorize the variational distribution $q(\alpha_i) \triangleq q(\mathbf{u}_i, \theta_i) = q(\mathbf{u}_i) q(\theta_i)$ and obtain

$$q(\alpha_i) = \mathcal{N}(\mathbf{u}_i | \mathbf{m}_{\mathbf{u}_i}, \Sigma_{\mathbf{u}_i}) \mathcal{N}(\theta_i | \mathbf{m}_{\theta_i}, \Sigma_{\theta_i}) \quad (10)$$

where $\Sigma_{\mathbf{u}_i} \triangleq \mathbf{C}_{\mathbf{u}_i} \mathbf{C}_{\mathbf{u}_i}^\top$ and $\Sigma_{\theta_i} \triangleq \mathbf{C}_{\theta_i} \mathbf{C}_{\theta_i}^\top$. Then, the gradient of the local ELBOs can be computed with respect to the variational parameters Φ_i using

$$\nabla_{\Phi_i} \mathcal{L}_i = \mathbb{E}_{q(\boldsymbol{\eta}_i)} \left[\nabla_{\Phi_i} \mathbb{E}_{p(\mathbf{f}_D | \alpha_i, k_i)} [\log p(\mathbf{y}_D | \mathbf{f}_D)] \right] - \nabla_{\Phi_i} \text{KL}[q(\alpha_i) || p(\alpha_i)] \quad (11)$$

The derivation of (11) can be obtained by applying the results in Appendix D of Hoang, Hoang, and Low (2017). Note that the reparameterization trick introduced above is used to make the first expectation operator $\mathbb{E}_{q(\boldsymbol{\eta}_i)}$ in (11) independent of the variational parameters Φ_i such that the gradient operator can be moved inside the first expectation w.r.t. $q(\boldsymbol{\eta}_i)$. Otherwise, if we assume (10) and optimize each \mathcal{L}_i in (9) directly w.r.t. the variational parameters $\Phi'_i \triangleq \text{vec}(\mathbf{m}_{\mathbf{u}_i}, \Sigma_{\mathbf{u}_i}, \mathbf{m}_{\theta_i}, \Sigma_{\theta_i})$, then the gradient operator cannot be moved into the first expectation in (9) since $\mathbb{E}_{q(\alpha_i)}$ depends on Φ'_i , which makes the estimation of the gradient $\nabla_{\Phi'_i} \mathcal{L}_i$ non-trivial.

³We use the widely-used standard multivariate Gaussian distribution as an example here. Similar to doubly stochastic VI (Titsias and Lázaro-Gredilla 2014), $q(\boldsymbol{\eta}_i)$ can be any standard continuous density function which yields a different $q(\alpha_i)$.

Given (11), the gradient of \mathcal{L}_i w.r.t. Φ_i can be approximated by sampling $\boldsymbol{\eta}_i \sim q(\boldsymbol{\eta}_i)$, as detailed in Appendix C. Unfortunately, the approximation of (11) is still computationally expensive for massive (e.g., million-sized) datasets since the estimation of $\nabla_{\Phi_i} \mathbb{E}_{p(\mathbf{f}_D | \boldsymbol{\alpha}_i, k_i)} [\log p(\mathbf{y}_D | \mathbf{f}_D)]$ incurs linear time in the data size $|\mathcal{D}|$ per SGA update. To resolve this issue, we exploit the *deterministic training conditional* (DTC) assumption of conditional independence among $f(\mathbf{x})$ for $\mathbf{x} \in \mathcal{D}$ given the inducing variables for deriving

$$\begin{aligned} & \nabla_{\Phi_i} \mathbb{E}_{p(\mathbf{f}_D | \boldsymbol{\alpha}_i, k_i)} [\log p(\mathbf{y}_D | \mathbf{f}_D)] \\ &= \nabla_{\Phi_i} \mathbb{E}_{p(\mathbf{f}_D | \boldsymbol{\alpha}_i, k_i)} [\sum_{\mathbf{x} \in \mathcal{D}} \log p(y(\mathbf{x}) | f(\mathbf{x}))] \\ &= \sum_{\mathbf{x} \in \mathcal{D}} \nabla_{\Phi_i} \mathbb{E}_{p(f(\mathbf{x}) | \boldsymbol{\alpha}_i, k_i)} [\log p(y(\mathbf{x}) | f(\mathbf{x}))]. \end{aligned} \quad (12)$$

Then, we can obtain an unbiased stochastic gradient estimate of \mathcal{L}_i w.r.t. Φ_i by uniformly sampling a *mini-batch* $\mathbf{y}_{\tilde{\mathcal{D}}}$ of the observed data where $\tilde{\mathcal{D}} \subseteq \mathcal{D}$ and the (fixed) batch size $|\tilde{\mathcal{D}}|$ is much smaller than $|\mathcal{D}|$. As a result, the SGA update of Φ_i incurs only constant time per iteration.

Optimizing $\mathcal{L}(q)$ w.r.t. $q(\mathbf{g})$. The optimization of $\mathcal{L}(q)$ can be performed by applying similar reparameterization trick and SGA. Specifically, let \mathcal{L}_i^* be the maximal local ELBO achieved using the above SGA and $\mathbf{g} \triangleq \mathbf{C}_g \boldsymbol{\eta}_g + \mathbf{m}_g$ with $q(\boldsymbol{\eta}_g) \triangleq \mathcal{N}(\boldsymbol{\eta}_g | \mathbf{0}, I)$. The gradient of $\mathcal{L}(q)$ can be derived in the same manner as that of (11):

$$\nabla_{\Phi_g} \mathcal{L} = \mathbb{E}_{q(\boldsymbol{\eta}_g)} \left[\sum_{i=1}^K \mathcal{L}_i^* \nabla_{\Phi_g} p(k_i | \mathbf{g}) \right] - \nabla_{\Phi_g} \text{KL} [q(\mathbf{g}) \| p(\mathbf{g})]$$

where $\Phi_g \triangleq \text{vec}(\mathbf{C}_g, \mathbf{m}_g)$. Then, the variational parameters Φ_g can also be updated via SGA by approximating the expectation operator in $\nabla_{\Phi_g} \mathcal{L}$ from sampling $\boldsymbol{\eta}_g \sim q(\boldsymbol{\eta}_g)$.

5 Kernel Posterior Belief and Predictive Belief of SGPR Model

The optimized variational parameters Φ_g^* and $\{\Phi_i^*\}_{i=1}^K$ can be computed from the above-mentioned stochastic optimization and used to induce the optimal variational distributions $q^*(\mathbf{g})$ and $q^*(\boldsymbol{\alpha}_i)$ for $i = 1, \dots, K$ via (10). Then, the posterior belief of the probabilistic kernel can be approximated by *Monte Carlo* (MC) sampling:

$$p(k | \mathbf{y}_D) \approx q^*(k) = \mathbb{E}_{q^*(\mathbf{g})} [p(k | \mathbf{g})] \approx \frac{1}{S} \sum_{s=1}^S p(k | \mathbf{g}^{(s)}) \quad (13)$$

where $\mathbf{g}^{(1)}, \dots, \mathbf{g}^{(S)}$ are i.i.d. samples from $q^*(\mathbf{g})$. Such an approximated kernel posterior belief is a discrete distribution where each $q^*(k_i) \in [0, 1]$ for $i = 1, \dots, K$ can be (a) easily interpreted to identify the ‘‘best’’ kernel whose posterior probability is much larger than that of the other kernels, and (b) exploited to avoid overconfident predictions by Bayesian model averaging, as will be shown in Section 6.

Recall from Section 3 that we approximate the posterior belief $p(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g} | \mathbf{y}_D)$ using the variational distribution for computing the predictive belief $p(f(\mathbf{x}_*) | \mathbf{y}_D)$. Given the optimal variational distributions $q^*(\mathbf{g})$ and $q^*(\boldsymbol{\alpha}_i)$ for

$i = 1, \dots, K$, the predictive belief in (3) can be approximated using

$$p(f(\mathbf{x}_*) | \mathbf{y}_D) \approx \sum_{i=1}^K q^*(k_i) \int p(f(\mathbf{x}_*) | \boldsymbol{\alpha}_i, k_i) q^*(\boldsymbol{\alpha}_i) d\boldsymbol{\alpha}_i \quad (14)$$

which yields the following approximated predictive mean and variance for $p(f(\mathbf{x}_*) | \mathbf{y}_D)$:

$$\begin{aligned} \mu_{\mathbf{x}_* | \mathcal{D}} &\approx \sum_{i=1}^K q^*(k_i) \mu_{(\mathbf{x}_*, i)} \\ \sigma_{\mathbf{x}_* | \mathcal{D}}^2 &\approx \sum_{i=1}^K q^*(k_i) (\sigma_{(\mathbf{x}_*, i)}^2 + \mu_{(\mathbf{x}_*, i)}^2) - \mu_{\mathbf{x}_* | \mathcal{D}}^2 \end{aligned} \quad (15)$$

where $\mu_{(\mathbf{x}_*, i)}$ and $\sigma_{(\mathbf{x}_*, i)}^2$ are the predictive mean and variance of $\int p(f(\mathbf{x}_*) | \boldsymbol{\alpha}_i, k_i) q^*(\boldsymbol{\alpha}_i) d\boldsymbol{\alpha}_i$ approximated by MC sampling. The derivations of (14), (15), and the steps for computing $\mu_{(\mathbf{x}_*, i)}$ and $\sigma_{(\mathbf{x}_*, i)}^2$ are in Appendix D.

As can be seen from (15), the time incurred to compute the predictive mean and variance is linear in the kernel size K , which is still expensive for GP prediction if K is large. To overcome this issue, we can consider constructing a much smaller kernel set $\tilde{\mathcal{K}} \subset \mathcal{K}$ including only the kernels with top-ranked posterior probabilities. Then, a new kernel posterior belief over $\tilde{\mathcal{K}}$ can be easily computed by optimizing a new ELBO $\tilde{\mathcal{L}}(q)$ constructed from reusing the optimized local ELBOs of the kernels in $\tilde{\mathcal{K}}$. As a result, the GP prediction can incur less time by pruning away kernels with low posterior probabilities while still maintaining the kernel uncertainty to avoid overconfident predictions.

In addition, one may notice that the GP predictive belief with the probabilistic kernel in (14) is in fact performing *Bayesian model averaging* (BMA) (Hoeting et al. 1999) over multiple GP models, each of which considers only one kernel k_i . Though it may seem straightforward to use BMA for GP prediction when multiple kernels are considered, our VBKS algorithm provides a principled way of doing so by deriving (14) from (3) using the VI framework and can scale the GP inference with the probabilistic kernel to big data, which is the main contribution of our work here.

6 Experiments and Discussion

This section empirically evaluates the performance of our VBKS algorithm on two small synthetic datasets and two massive real-world datasets. The kernel posterior belief is computed using (13) with $S = 2000$. The real-world experiments are performed on a Linux system with 5 Nvidia GeForce GTX 1080 GPUs. Stochastic optimization for VBKS is performed in a distributed manner over the 5 GPUs using GPflow (Matthews et al. 2017).

6.1 Synthetic Experiments

We will first demonstrate the performance of our VBKS algorithm in identifying kernel(s) that can capture the underlying correlation structure of two synthetic datasets. These two synthetic datasets are generated using the respective true composite kernels $\text{PER} \times \text{LIN} \times \text{RQ}$ and $(\text{PER} + \text{RQ}) \times \text{LIN}$ with fixed hyperparameters (Appendix E.1). We set the input dimension as $d \triangleq 1$ and input domain as $\mathcal{X} \triangleq [-10, 10]$. A

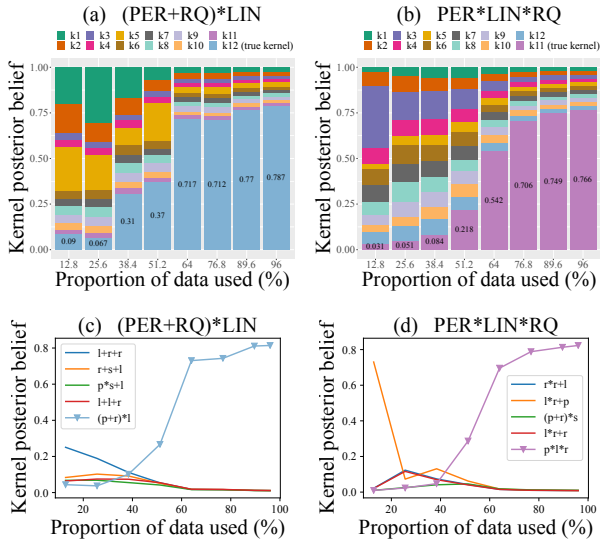


Figure 2: Graphs of kernel posterior belief achieved by VBKS vs. proportion of data used in stochastic optimization for two synthetic datasets with (a-b) kernel set \mathcal{K}_{12} and (c-d) kernel set \mathcal{K}_{144} . The ‘s’, ‘r’, ‘p’, and ‘l’ in the legend denote SE, RQ, PER, and LIN kernels, respectively.

set \mathcal{D}_0 of 256 inputs are randomly sampled from \mathcal{X} and their corresponding outputs $\mathbf{y}_{\mathcal{D}_0}$ are sampled from a full-rank GP prior. Then, we sample another set \mathcal{D}_1 of 1000 inputs from \mathcal{X} , compute their predictive mean $\mu_{\mathcal{D}_1|\mathcal{D}_0} \triangleq (\mu_{\mathbf{x}_*|\mathcal{D}_0})_{\mathbf{x}_* \in \mathcal{D}_1}$ via (1), and use $(\mathcal{D}_1, \mu_{\mathcal{D}_1|\mathcal{D}_0})$ as the synthetic dataset. In all the synthetic experiments, we use $|\mathcal{U}| = 16$ inducing inputs and a batch size $|\tilde{\mathcal{D}}| = 32$ to perform the SGA update per iteration. Two kernel sets $|\mathcal{K}_{12}| = 12$ and $|\mathcal{K}_{144}| = 144$ are used to evaluate the performance of our VBKS algorithm where \mathcal{K}_{144} contains kernels constructed from the base kernels (i.e., SE, RQ, LIN, PER) by applying the grammar rules of Duvenaud et al. (2013) until level 3 while \mathcal{K}_{12} includes 10 kernels randomly sampled from \mathcal{K}_{144} and the two true kernels, as shown in Appendix E.1. The small kernel set \mathcal{K}_{12} is constructed such that the posterior probabilities of all the kernels in the kernel space can be easily observed and visualized.

Figs. 2a and 2b show the kernel posterior belief over \mathcal{K}_{12} that is produced by our VBKS algorithm. Figs. 2c and 2d include only the kernels whose posterior probabilities have ever been ranked as the top two among \mathcal{K}_{144} during stochastic optimization for VBKS. It can be observed in all the experiments that the posterior probability of the true kernel is small at the beginning and, with a growing data size, is increased by our VBKS algorithm to be around 0.8 which is much larger than that of the other kernels. Though the SGPR model used by our VBKS algorithm produces a low-rank approximation of the true covariance structure of the data using a small set of inducing variables, our algorithm can find the kernel that correctly interprets the underlying correlation structure. It can be observed from Figs. 2a and 2b that the kernel uncertainty is large (i.e., no kernel has a much

larger posterior probability than the others) when less than half of the data is used in these experiments. In such cases, the high kernel uncertainty shows that the current observed data is insufficient in identifying a single kernel that fits the true correlation structure much better than the other kernels. This implies the need to collect more data (as has been done in the experiments) or perform GP prediction with the kernel uncertainty.

6.2 Real-World Experiments

This section empirically evaluates the performance and time efficiency of our VBKS algorithm on two massive real-world datasets: (a) Swissgrid dataset⁴ contains 210,336 records of the total energy consumed by end users in the Swiss control block from January 1, 2009 to December 31, 2015 in every 15 minutes, and (b) *indoor environmental quality* (IEQ) dataset⁵ contains temperature ($^{\circ}\text{C}$) taken in every 31 seconds between February 28 and April 5, 2004 by 54 sensors deployed in the Intel Berkeley Research lab.

The candidate kernel set \mathcal{K}_{144} is used in the experiments for both datasets. The predictive performance of our VBKS algorithm is obtained using BMA over 10 kernels with top-ranked posterior probabilities and compared with that of (a) *Bayesian optimization* (BO) for kernel selection (Malkomes and Garnett 2015) which automatically finds the kernel with the highest model evidence using BO, (b) random algorithm which randomly selects a kernel from \mathcal{K}_{144} at the beginning of each experiment, and (c) VBKS performing GP prediction with a single kernel that yields the highest posterior probability (VBKS-s). For each dataset, 20,000 observations are randomly selected to form the test set \mathcal{T} . The *root mean squared error* (RMSE) metric $\sqrt{|\mathcal{T}|^{-1} \sum_{\mathbf{x}_* \in \mathcal{T}} (y(\mathbf{x}_*) - \mu_{\mathbf{x}_*|\mathcal{D}})^2}$ is used to evaluate the predictive performance of the tested algorithms. The RMSEs of the VBKS(-s) and random algorithms are averaged over 10 and 50 independent runs, respectively. The error bars are computed in the form of standard deviation. To save some time from sampling $q(\theta_i)$, the hyperparameters used by all tested algorithms are point estimates by setting \mathbf{C}_{θ_i} as a zero matrix.

Swissgrid Dataset. We use $|\mathcal{U}| = 800$ inducing inputs and a batch size $|\tilde{\mathcal{D}}| = 128$ for SGA update per iteration. The kernel posterior belief produced by our VBKS algorithm is evaluated after every 1.28% of data are used, as shown in Fig. 3a. For clarity, we only visualize the kernels whose posterior probabilities have ever been ranked as the top three during stochastic optimization for VBKS. It can be observed that a group of three kernels (rather than a single kernel) achieve comparable posterior probabilities which are larger than that of the other kernels. This implies that the uncertainty among the kernels need to be considered when the selected kernels are used to interpret the correlation structure or perform GP prediction, which is a key benefit of using our VBKS algorithm. The predictive performance of the tested algorithms for Swissgrid dataset is shown in Fig. 3b:

⁴<https://www.swissgrid.ch>

⁵<http://db.csail.mit.edu/labdata/labdata.html>

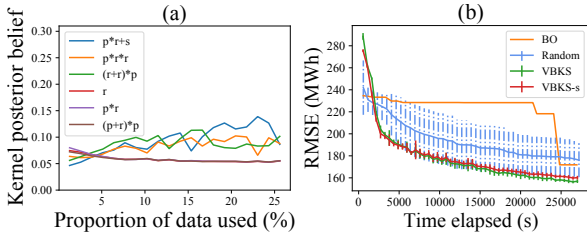


Figure 3: Graphs of (a) kernel posterior belief over selected kernels that is produced by our VBKS algorithm in one run vs. proportion of data used in stochastic optimization, and (b) RMSE vs. time incurred by tested algorithms for Swissgrid dataset.

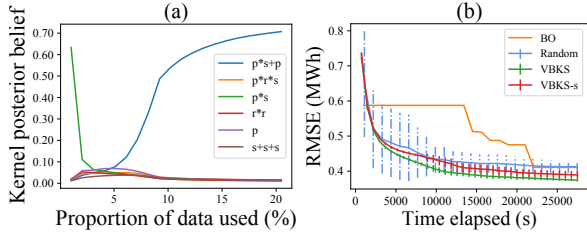


Figure 4: Graphs of (a) kernel posterior belief over selected kernels that is produced by our VBKS algorithm in one run vs. proportion of data used in stochastic optimization, and (b) RMSE vs. time incurred by tested algorithms for IEQ dataset.

Both the VBKS and VBKS-s algorithms converge faster to a smaller RMSE than the other tested algorithms. VBKS performs better than VBKS-s since BMA can benefit from different kernels in modeling the data when no kernel truly stands out. The BO algorithm performs poorly because it has to approximate the distance between kernels using a small subset of data (i.e., of size 200 here).⁶ If the small subset of data is not large enough to approximate the kernel distances well, the BO performance will degrade, which is the case in our experiments.

IEQ Dataset. In this experiment, the time and locations of the sensors for producing the temperature readings are used jointly as the input (i.e., $d = 3$). The first 1 million valid data points are selected for our experiments. We use $|\mathcal{U}| = 1000$ inducing inputs and a batch size $|\tilde{\mathcal{D}}| = 512$ for SGA update per iteration. Fig. 4 shows the kernel posterior belief and RMSE of the tested algorithms for IEQ dataset. Different from the results in Fig. 3a for Swissgrid dataset, Fig. 4a shows that the posterior probability of a specific kernel (i.e., $\text{PER} \times \text{SE} + \text{PER}$) increases fast to be much larger than that of the other kernels. However, it can be observed from Fig. 4b that VBKS still outperforms VBKS-s because the training of GP model using $\text{PER} \times \text{SE} + \text{PER}$ overfits to the training data and our VBKS algorithm with BMA helps to reduce the

⁶We also tried a larger subset of data of size 500 to better approximate the kernel distances in BO, which yields similar predictive performance but incurs much more time.

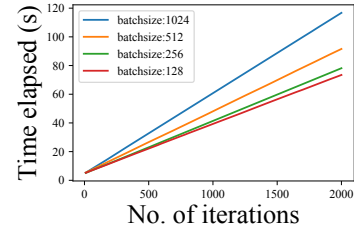


Figure 5: Graph of total incurred time of our VBKS algorithm vs. number of iterations of SGA updates with $|\mathcal{U}| = 800$ and varying batch sizes $|\tilde{\mathcal{D}}|$.

overfitting. VBKS also converges to a smaller RMSE than all other tested algorithms, as shown in Fig. 4b. In addition, for both Swissgrid and IEQ datasets, we observe that VBKS has consistently achieved smaller RMSE than VBKS-s (albeit slightly) in all 10 independent runs, which demonstrates the benefit of applying BMA. The results of all these 10 runs (instead of the averaged RMSE) are in Appendix E.2.

Scalability. Fig. 5 shows the time efficiency of our VBKS algorithm for different batch sizes for SGA update per iteration. As expected, the total time incurred by VBKS increases linearly in the number of iterations of SGA updates.

7 Conclusion

This paper describes a novel VBKS algorithm for SGPR models that considers a probabilistic kernel and exploits the kernel uncertainty to avoid overconfident predictions. A stochastic optimization method for VBKS is proposed for learning the kernel variational distribution together with the other variational variables (i.e., inducing variables and kernel hyperparameters). Our VBKS algorithm achieves scalability in the kernel size K by decomposing the variational lower bound into an additive form such that each additive term (i.e., local ELBO) depends on the variational variables of only one kernel and can thus be maximized independently. Then, each additive local ELBO is optimized via SGA by sampling from both the variational distribution and the data, which scales to big data since it incurs constant time per iteration of SGA update. The predictive performance of our VBKS algorithm with BMA is shown to outperform the other tested kernel selection algorithms. A limitation of VBKS is that it uses a finite and fixed kernel space, which does not allow flexible exploration. In our future work, we will consider expanding the kernel space during stochastic optimization according to the intermediate learning outcomes and extending VBKS to operate with an infinite kernel space and a deep GP model (Yu et al. 2019a).

Acknowledgments. This research is supported by the National Research Foundation, Prime Minister’s Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) program, Singapore-MIT Alliance for Research and Technology (SMART) Future Urban Mobility (FM) IRG, the Singapore Ministry of Education Academic Research Fund Tier 2, MOE2016-T2-2-156, and

the Joint Funds of the National Natural Science Foundation of China under Key Program Grant U1713212.

References

- Benton, G. W.; Maddox, W. J.; Salkey, J. P.; Albinati, J.; and Wilson, A. G. 2019. Function-space distributions over kernels. In *Proc. NeurIPS*.
- Chen, J.; Low, K. H.; Tan, C. K.-Y.; Oran, A.; Jaillet, P.; Dolan, J. M.; and Sukhatme, G. S. 2012. Decentralized data fusion and active sensing with mobile sensors for modeling and predicting spatiotemporal traffic phenomena. In *Proc. UAI*, 163–173.
- Chen, J.; Cao, N.; Low, K. H.; Ouyang, R.; Tan, C. K.-Y.; and Jaillet, P. 2013. Parallel Gaussian process regression with low-rank covariance matrix approximations. In *Proc. UAI*, 152–161.
- Chen, J.; Low, K. H.; Jaillet, P.; and Yao, Y. 2015. Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems. *IEEE Transactions on Automation Science and Engineering* 12(3):901–921.
- Duvenaud, D.; Lloyd, J. R.; Grosse, R.; Tenenbaum, J. B.; and Ghahramani, Z. 2013. Structure discovery in nonparametric regression through compositional kernel search. In *Proc. ICML*, 1166–1174.
- Gal, Y., and Turner, R. 2015. Improving the Gaussian process sparse spectrum approximation by representing uncertainty in frequency inputs. In *Proc. ICML*, 655–664.
- Hensman, J.; Matthews, A. G.; Filippone, M.; and Ghahramani, Z. 2015. MCMC for variationally sparse Gaussian processes. In *Proc. NeurIPS*, 1648–1656.
- Hoang, T. N.; Hoang, Q. M.; and Low, K. H. 2015. A unifying framework of anytime sparse Gaussian process regression models with stochastic variational inference for big data. In *Proc. ICML*, 569–578.
- Hoang, T. N.; Hoang, Q. M.; and Low, K. H. 2016. A distributed variational inference framework for unifying parallel sparse Gaussian process regression models. In *Proc. ICML*, 382–391.
- Hoang, Q. M.; Hoang, T. N.; and Low, K. H. 2017. A generalized stochastic variational Bayesian hyperparameter learning framework for sparse spectrum Gaussian process regression. In *Proc. AAAI*, 2007–2014.
- Hoeting, J. A.; Madigan, D.; Raftery, A. E.; and Volinsky, C. T. 1999. Bayesian model averaging: a tutorial. *Statistical science* 382–401.
- Kim, H., and Teh, Y. W. 2018. Scaling up the automatic statistician: Scalable structure discovery using Gaussian processes. In *Proc. AISTATS*.
- Lázaro-Gredilla, M.; Quiñero-Candela, J.; Rasmussen, C. E.; and Figueiras-Vidal, A. R. 2010. Sparse spectrum Gaussian process regression. *JMLR* 11:1865–1881.
- Lloyd, J. R.; Duvenaud, D.; Grosse, R.; Tenenbaum, J. B.; and Ghahramani, Z. 2014. Automatic construction and natural-language description of nonparametric regression models. In *Proc. AAAI*.
- Low, K. H.; Chen, J.; Hoang, T. N.; Xu, N.; and Jaillet, P. 2015a. Recent advances in scaling up Gaussian process predictive models for large spatiotemporal data. In Ravela, S., and Sandu, A., eds., *Proc. Dynamic Data-driven Environmental Systems Science Conference (DyDESS'14)*. LNCS 8964, Springer.
- Low, K. H.; Yu, J.; Chen, J.; and Jaillet, P. 2015b. Parallel Gaussian process regression for big data: Low-rank representation meets Markov approximation. In *Proc. AAAI*.
- Lu, X.; Gonzalez, J.; Dai, Z.; and Lawrence, N. 2018. Structured variationally auto-encoded optimization. In *Proc. ICML*, 3273–3281.
- Malkomes, G., and Garnett, R. 2015. Active structure discovery for Gaussian processes. In *ICML Workshop on AutoML*.
- Malkomes, G.; Schaff, C.; and Garnett, R. 2016. Bayesian optimization for automated model selection. In *Proc. NeurIPS*, 2900–2908.
- Matthews, A. G. d. G.; van der Wilk, M.; Nickson, T.; Fujii, K.; Boukouvalas, A.; León-Villagrà, P.; Ghahramani, Z.; and Hensman, J. 2017. GPflow: A Gaussian process library using TensorFlow. *JMLR* 18(40):1–6.
- Oliva, J. B.; Dubey, A.; Wilson, A. G.; Póczos, B.; Schneider, J.; and Xing, E. P. 2016. Bayesian nonparametric kernel-learning. In *Proc. AISTATS*, 1078–1086.
- Quiñero-Candela, J., and Rasmussen, C. E. 2005. A unifying view of sparse approximate Gaussian process regression. *JMLR* 6:1939–1959.
- Rasmussen, C. E., and Williams, C. K. 2006. *Gaussian processes for machine learning*. MIT Press.
- Titsias, M. K., and Lázaro-Gredilla, M. 2013. Variational inference for Mahalanobis distance metrics in Gaussian process regression. In *Proc. NeurIPS*, 279–287.
- Titsias, M. K., and Lázaro-Gredilla, M. 2014. Doubly stochastic variational Bayes for non-conjugate inference. In *Proc. ICML*, 1971–1979.
- Titsias, M. K. 2009. Variational learning of inducing variables in sparse Gaussian processes. In *Proc. AISTATS*.
- Wilson, A. G., and Adams, R. P. 2013. Gaussian process kernels for pattern discovery and extrapolation. In *Proc. ICML*.
- Xu, N.; Low, K. H.; Chen, J.; Lim, K. K.; and Özgül, E. B. 2014. GP-Localize: Persistent mobile robot localization using online sparse Gaussian process observation model. In *Proc. AAAI*, 2585–2592.
- Yu, H.; Chen, Y.; Dai, Z.; Low, K. H.; and Jaillet, P. 2019a. Implicit posterior variational inference for deep Gaussian processes. In *Proc. NeurIPS*.
- Yu, H.; Hoang, T. N.; Low, K. H.; and Jaillet, P. 2019b. Stochastic variational inference for Bayesian sparse Gaussian process regression. In *Proc. IJCNN*.

A Derivation of (5)

Since $p(\mathbf{y}_D) = p(\mathbf{y}_D, \mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})/p(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g}|\mathbf{y}_D)$ for any $\mathbf{f}_D, \boldsymbol{\alpha}, k$, and \mathbf{g} ,

$$\log p(\mathbf{y}_D) = \log \frac{p(\mathbf{y}_D, \mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})}{p(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g}|\mathbf{y}_D)}. \quad (16)$$

Let $q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})$ be an arbitrary probability density function. Then, we can take an expectation w.r.t. $q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})$ on both side of (16), which yields

$$\begin{aligned} \log p(\mathbf{y}_D) &= \mathbb{E}_{q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})} \left[\log \frac{p(\mathbf{y}_D, \mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})}{p(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g}|\mathbf{y}_D)} \right] \\ &= \mathbb{E}_{q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})} \left[\log \left(\frac{p(\mathbf{y}_D, \mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})}{q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})} \frac{q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})}{p(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g}|\mathbf{y}_D)} \right) \right] \\ &= \mathbb{E}_{q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})} \left[\log \frac{p(\mathbf{y}_D|\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})p(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})}{q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})} \right] \\ &\quad + \mathbb{E}_{q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})} \left[\log \frac{q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})}{p(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g}|\mathbf{y}_D)} \right] \\ &= \mathbb{E}_{q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})} [\log p(\mathbf{y}_D|\mathbf{f}_D)] \\ &\quad - \mathbb{E}_{q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})} \left[\log \frac{q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})}{p(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})} \right] \\ &\quad + \mathbb{E}_{q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})} \left[\log \frac{q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})}{p(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g}|\mathbf{y}_D)} \right] \\ &= \mathbb{E}_{q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})} [\log p(\mathbf{y}_D|\mathbf{f}_D)] \\ &\quad - \text{KL} [q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})||p(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})] \\ &\quad + \text{KL} [q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})||p(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g}|\mathbf{y}_D)] \\ &= \mathcal{L}(q) + \text{KL} [q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})||p(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g}|\mathbf{y}_D)] \end{aligned}$$

where the fourth equality is due to the fact that \mathbf{y}_D is conditionally independent of $(\boldsymbol{\alpha}, k, \mathbf{g})$ given \mathbf{f}_D since $y(\mathbf{x}) \triangleq f(\mathbf{x}) + \epsilon$ and the fifth equality follows from the definition of KL divergence.

B Derivation of (8)

From (5),

$$\begin{aligned} \mathcal{L}(q) &= \mathbb{E}_{q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})} [\log p(\mathbf{y}_D|\mathbf{f}_D)] \\ &\quad - \text{KL} [q(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})||p(\mathbf{f}_D, \boldsymbol{\alpha}, k, \mathbf{g})] \\ &= \mathbb{E}_{p(k|\mathbf{g})q(\mathbf{g})} [\mathbb{E}_{p(\mathbf{f}_D|\boldsymbol{\alpha}, k)q(\boldsymbol{\alpha})} [\log p(\mathbf{y}_D|\mathbf{f}_D)]] \\ &\quad - \sum_{i=1}^K \text{KL} [q(\boldsymbol{\alpha}_i)||p(\boldsymbol{\alpha}_i)] - \text{KL} [q(\mathbf{g})||p(\mathbf{g})] \\ &= \mathbb{E}_{q(\mathbf{g})} \left[\sum_{i=1}^K p(k_i|\mathbf{g}) \mathbb{E}_{p(\mathbf{f}_D|\boldsymbol{\alpha}_i, k_i)q(\boldsymbol{\alpha}_i)} [\log p(\mathbf{y}_D|\mathbf{f}_D)] \right] \\ &\quad - \sum_{i=1}^K \text{KL} [q(\boldsymbol{\alpha}_i)||p(\boldsymbol{\alpha}_i)] - \text{KL} [q(\mathbf{g})||p(\mathbf{g})] \\ &= \mathbb{E}_{q(\mathbf{g})} \left[\sum_{i=1}^K p(k_i|\mathbf{g}) \mathbb{E}_{p(\mathbf{f}_D|\boldsymbol{\alpha}_i, k_i)q(\boldsymbol{\alpha}_i)} [\log p(\mathbf{y}_D|\mathbf{f}_D)] \right] \\ &\quad - \mathbb{E}_{q(\mathbf{g})} \left[\sum_{i=1}^K \text{KL} [q(\boldsymbol{\alpha}_i)||p(\boldsymbol{\alpha}_i)] \right] - \text{KL} [q(\mathbf{g})||p(\mathbf{g})] \\ &= \mathbb{E}_{q(\mathbf{g})} \left[\sum_{i=1}^K p(k_i|\mathbf{g}) \mathcal{L}_i(q) \right] - \text{KL} [q(\mathbf{g})||p(\mathbf{g})] \end{aligned}$$

where the second equality is due to (6), (7), and the additive property of KL divergence for independent distributions, and the fourth equality follows from the fact that \mathbf{f}_D is conditionally independent of $\boldsymbol{\alpha}_j$ for $j \neq i$ given kernel k_i and $\boldsymbol{\alpha}_i$.

C Derivation of (11)

C.1 Derivation of $\mathbb{E}_{p(\mathbf{f}_D|\boldsymbol{\alpha}_i, k_i)} [\log p(\mathbf{y}_D|\mathbf{f}_D)]$

Firstly, $p(\mathbf{f}_D|\boldsymbol{\alpha}_i, k_i) = p(\mathbf{f}_D|\mathbf{u}_i, \boldsymbol{\theta}_i, k_i)$ is a Gaussian with the following mean and covariance:

$$\begin{aligned} \boldsymbol{\mu}_{\mathcal{D}|\boldsymbol{\alpha}_i} &\triangleq \Sigma_{\langle \mathcal{D}, i \rangle \langle \mathcal{U}, i \rangle}^{\boldsymbol{\theta}_i} (\Sigma_{\langle \mathcal{U}, i \rangle \langle \mathcal{U}, i \rangle}^{\boldsymbol{\theta}_i})^{-1} \mathbf{u}_i \\ \Sigma_{\mathcal{D}|\boldsymbol{\alpha}_i} &\triangleq \Sigma_{\langle \mathcal{D}, i \rangle \langle \mathcal{D}, i \rangle}^{\boldsymbol{\theta}_i} - \Sigma_{\langle \mathcal{D}, i \rangle \langle \mathcal{U}, i \rangle}^{\boldsymbol{\theta}_i} (\Sigma_{\langle \mathcal{U}, i \rangle \langle \mathcal{U}, i \rangle}^{\boldsymbol{\theta}_i})^{-1} \Sigma_{\langle \mathcal{U}, i \rangle \langle \mathcal{D}, i \rangle}^{\boldsymbol{\theta}_i} \end{aligned}$$

where $\Sigma_{\langle \mathcal{D}, i \rangle \langle \mathcal{U}, i \rangle}^{\boldsymbol{\theta}_i} \triangleq (k_i^{\boldsymbol{\theta}_i}(\mathbf{x}, \mathbf{x}'))_{\mathbf{x} \in \mathcal{X}, \mathbf{x}' \in \mathcal{U}}$, $\Sigma_{\langle \mathcal{U}, i \rangle \langle \mathcal{U}, i \rangle}^{\boldsymbol{\theta}_i} \triangleq (k_i^{\boldsymbol{\theta}_i}(\mathbf{x}, \mathbf{x}'))_{\mathbf{x}, \mathbf{x}' \in \mathcal{U}}$, and $k_i^{\boldsymbol{\theta}_i}(\mathbf{x}, \mathbf{x}')$ denotes $k_i(\mathbf{x}, \mathbf{x}')$ com-

puted using hyperparameters θ_i . Then,

$$\begin{aligned}
& \mathbb{E}_{p(\mathbf{f}_D|\alpha_i, k_i)}[\log p(\mathbf{y}_D|\mathbf{f}_D)] \\
&= \int p(\mathbf{f}_D|\alpha_i, k_i) \log \mathcal{N}(\mathbf{y}_D|\mathbf{f}_D, \sigma_n^2 I) d\mathbf{f}_D \\
&= \int p(\mathbf{f}_D|\alpha_i, k_i) \left(-\frac{|\mathcal{D}|}{2} \log(2\pi\sigma_n^2) \right. \\
&\quad \left. -\frac{1}{2\sigma_n^2} (\mathbf{y}_D^\top \mathbf{y}_D - 2\mathbf{f}_D^\top \mathbf{y}_D + \mathbf{f}_D^\top \mathbf{f}_D) \right) d\mathbf{f}_D \\
&= -\frac{|\mathcal{D}|}{2} \log(2\pi\sigma_n^2) - \frac{1}{2\sigma_n^2} (\mathbf{y}_D^\top \mathbf{y}_D) \\
&\quad + \frac{1}{\sigma_n^2} \mathbb{E}_{p(\mathbf{f}_D|\alpha_i, k_i)}[\mathbf{y}_D^\top \mathbf{f}_D] - \frac{1}{2\sigma_n^2} \mathbb{E}_{p(\mathbf{f}_D|\alpha_i, k_i)}[\mathbf{f}_D^\top \mathbf{f}_D] \\
&= -\frac{|\mathcal{D}|}{2} \log(2\pi\sigma_n^2) - \frac{1}{2\sigma_n^2} (\mathbf{y}_D^\top \mathbf{y}_D) \\
&\quad + \frac{1}{2\sigma_n^2} \left(2\mathbf{y}_D^\top \boldsymbol{\mu}_D|\alpha_i - \boldsymbol{\mu}_D^\top|\alpha_i \boldsymbol{\mu}_D|\alpha_i - \text{tr}(\Sigma_D|\alpha_i) \right).
\end{aligned}$$

The gradient $\nabla_{\Phi_i} \mathbb{E}_{p(\mathbf{f}_D|\alpha_i, k_i)}[\log p(\mathbf{y}_D|\mathbf{f}_D)]$ can thus be computed automatically using TensorFlow based on the above analytic equation.

C.2 Derivation of $\text{KL}[q(\alpha_i)||p(\alpha_i)]$

Using the definition of KL divergence,

$$\begin{aligned}
& \text{KL}[q(\alpha_i)||p(\alpha_i)] \\
&= \text{KL}[q(\mathbf{u}_i)q(\theta_i)||p(\mathbf{u}_i|\theta_i)p(\theta_i)] \\
&= \mathbb{E}_{q(\mathbf{u}_i)q(\theta_i)} \left[\log \frac{q(\mathbf{u}_i)q(\theta_i)}{p(\mathbf{u}_i|\theta_i)p(\theta_i)} \right] \\
&= \mathbb{E}_{q(\mathbf{u}_i)q(\theta_i)} \left[\log q(\mathbf{u}_i) + \log \frac{q(\theta_i)}{p(\theta_i)} - \log p(\mathbf{u}_i|\theta_i) \right] \\
&= -\mathbb{H}[q(\mathbf{u}_i)] + \text{KL}[q(\theta_i)||p(\theta_i)] - \mathbb{E}_{q(\mathbf{u}_i)q(\theta_i)}[\log p(\mathbf{u}_i|\theta_i)]
\end{aligned}$$

where the entropy and the KL divergence terms can be computed analytically such that their gradients w.r.t. Φ_i can be computed automatically using TensorFlow. Though the last term cannot be computed analytically, we can approximate its gradient w.r.t. Φ_i via the reparameterization trick (Section 4.1) by drawing i.i.d. samples $\boldsymbol{\eta}_i^{(s)}$ for $s = 1, \dots, S$ from $q(\boldsymbol{\eta}_i)$: $\nabla_{\Phi_i} \mathbb{E}_{q(\mathbf{u}_i)q(\theta_i)}[\log p(\mathbf{u}_i|\theta_i)] = \nabla_{\Phi_i} \mathbb{E}_{q(\boldsymbol{\eta}_i)}[\log p(\mathbf{u}_i|\theta_i)] = \mathbb{E}_{q(\boldsymbol{\eta}_i)}[\nabla_{\Phi_i} \log p(\mathbf{u}_i|\theta_i)] \approx (1/S) \sum_{s=1}^S \nabla_{\Phi_i} \log p(\mathbf{u}_i^{(s)}|\theta_i^{(s)})$.

D Derivation of the Approximated Predictive Belief

D.1 Derivation of (14)

From (3),

$$\begin{aligned}
& p(f(\mathbf{x}_*)|\mathbf{y}_D) \\
&= \sum_{i=1}^K \int p(f(\mathbf{x}_*)|\mathbf{f}_D, \alpha, k_i, \mathbf{g}) p(\mathbf{f}_D, \alpha, k_i, \mathbf{g}|\mathbf{y}_D) d\mathbf{f}_D d\alpha d\mathbf{g} \\
&\approx \sum_{i=1}^K \int p(f(\mathbf{x}_*)|\mathbf{f}_D, \alpha, k_i, \mathbf{g}) p(\mathbf{f}_D|\alpha, k_i) \\
&\quad p(k_i|\mathbf{g}) q^*(\mathbf{g}) \prod_{j=1}^K q^*(\alpha_j) d\mathbf{f}_D d\alpha d\mathbf{g} \\
&= \sum_{i=1}^K \int p(f(\mathbf{x}_*)|\alpha_i, k_i) \left(\int p(\mathbf{f}_D|\alpha, k_i) d\mathbf{f}_D \right) \\
&\quad \left(\int p(k_i|\mathbf{g}) q^*(\mathbf{g}) d\mathbf{g} \right) \left(\prod_{j \neq i} \int q^*(\alpha_j) d\alpha_j \right) q^*(\alpha_i) d\alpha_i \\
&= \sum_{i=1}^K q^*(k_i) \int p(f(\mathbf{x}_*)|\alpha_i, k_i) q^*(\alpha_i) d\alpha_i
\end{aligned}$$

where the approximation is due to (7), the second equality follows from the DTC assumption that (a) $f(\mathbf{x}_*)$ is conditionally independent of \mathbf{f}_D given the inducing variables \mathbf{u}_i , and the assumption that (b) $f(\mathbf{x}_*)$ is conditionally independent of \mathbf{g} and α_j for $j \neq i$ given kernel k_i and α_i , and the last equality is due to (13).

D.2 Derivation of (15)

In this subsection, we will first introduce how to approximate the predictive mean and variance of each $\int p(f(\mathbf{x}_*)|\alpha_i, k_i) q^*(\alpha_i) d\alpha_i$ for $i = 1, \dots, K$.

Following the test conditional in (Quiñero-Candela and Rasmussen 2005), $p(f(\mathbf{x}_*)|\mathbf{u}_i, \theta_i, k_i)$ is a Gaussian with the following mean and variance:

$$\begin{aligned}
\mu_{\langle \mathbf{x}_*, i \rangle | \alpha_i} &\triangleq \Sigma_{\langle \mathbf{x}_*, i \rangle \langle \mathcal{U}, i \rangle}^{\theta_i} (\Sigma_{\langle \mathcal{U}, i \rangle \langle \mathcal{U}, i \rangle}^{\theta_i})^{-1} \mathbf{u}_i \\
\sigma_{\langle \mathbf{x}_*, i \rangle | \alpha_i}^2 &\triangleq k_i^{\theta_i}(\mathbf{x}_*, \mathbf{x}_*) - \Sigma_{\langle \mathbf{x}_*, i \rangle \langle \mathcal{U}, i \rangle}^{\theta_i} (\Sigma_{\langle \mathcal{U}, i \rangle \langle \mathcal{U}, i \rangle}^{\theta_i})^{-1} \Sigma_{\langle \mathcal{U}, i \rangle \langle \mathbf{x}_*, i \rangle}^{\theta_i}
\end{aligned}$$

where $\Sigma_{\langle \mathbf{x}_*, i \rangle \langle \mathcal{U}, i \rangle}^{\theta_i} \triangleq (k_i^{\theta_i}(\mathbf{x}_*, \mathbf{x}))_{\mathbf{x} \in \mathcal{U}}$, $\Sigma_{\langle \mathcal{U}, i \rangle \langle \mathcal{U}, i \rangle}^{\theta_i} \triangleq (k_i^{\theta_i}(\mathbf{x}, \mathbf{x}'))_{\mathbf{x}, \mathbf{x}' \in \mathcal{U}}$, $\Sigma_{\langle \mathcal{U}, i \rangle \langle \mathbf{x}_*, i \rangle}^{\theta_i} \triangleq (\Sigma_{\langle \mathbf{x}_*, i \rangle \langle \mathcal{U}, i \rangle}^{\theta_i})^\top$, and $k_i^{\theta_i}(\mathbf{x}, \mathbf{x}')$ denotes $k_i(\mathbf{x}, \mathbf{x}')$ computed using hyperparameters θ_i . Then,

$$\begin{aligned}
& \int p(f(\mathbf{x}_*)|\alpha_i, k_i) q^*(\alpha_i) d\alpha_i \\
&= \int \left(\int p(f(\mathbf{x}_*)|\mathbf{u}_i, \theta_i, k_i) q^*(\mathbf{u}_i) d\mathbf{u}_i \right) q^*(\theta_i) d\theta_i
\end{aligned} \tag{17}$$

where the inner integration $\int p(f(\mathbf{x}_*)|\mathbf{u}_i, \theta_i, k_i) q^*(\mathbf{u}_i) d\mathbf{u}_i$ can be computed analytically as a Gaussian with the follow-

ing mean and variance:

$$\begin{aligned}\mu_{\langle \mathbf{x}_*, i \rangle | \theta_i} &\triangleq \Sigma_{\langle \mathbf{x}_*, i \rangle \langle \mathcal{U}, i \rangle}^{\theta_i} (\Sigma_{\langle \mathcal{U}, i \rangle \langle \mathcal{U}, i \rangle}^{\theta_i})^{-1} \mathbf{m}_{\mathbf{u}_i}^* \\ \sigma_{\langle \mathbf{x}_*, i \rangle | \theta_i}^2 &\triangleq k_i(\mathbf{x}_*, \mathbf{x}_*) - \Sigma_{\langle \mathbf{x}_*, i \rangle \langle \mathcal{U}, i \rangle}^{\theta_i} (\Sigma_{\langle \mathcal{U}, i \rangle \langle \mathcal{U}, i \rangle}^{\theta_i})^{-1} \Sigma_{\langle \mathcal{U}, i \rangle \langle \mathbf{x}_*, i \rangle}^{\theta_i} \\ &+ \Sigma_{\langle \mathbf{x}_*, i \rangle \langle \mathcal{U}, i \rangle}^{\theta_i} (\Sigma_{\langle \mathcal{U}, i \rangle \langle \mathcal{U}, i \rangle}^{\theta_i})^{-1} \Sigma_{\mathbf{u}_i}^* (\Sigma_{\langle \mathcal{U}, i \rangle \langle \mathcal{U}, i \rangle}^{\theta_i})^{-1} \Sigma_{\langle \mathcal{U}, i \rangle \langle \mathbf{x}_*, i \rangle}^{\theta_i}.\end{aligned}$$

Though the outer integration w.r.t. θ_i cannot be computed analytically for all kernel types (except for some simple kernels such as SE and LIN), it can be approximated by drawing i.i.d. samples $\theta_i^{(1)}, \dots, \theta_i^{(S)}$ from $q^*(\theta_i)$, which yields the following predictive mean and variance for (17):

$$\begin{aligned}\mu_{\langle \mathbf{x}_*, i \rangle} &\approx \frac{1}{S} \sum_{s=1}^S \mu_{\langle \mathbf{x}_*, i \rangle | \theta_i^{(s)}} \\ \sigma_{\langle \mathbf{x}_*, i \rangle}^2 &\approx \frac{1}{S} \sum_{s=1}^S (\sigma_{\langle \mathbf{x}_*, i \rangle | \theta_i^{(s)}}^2 + \mu_{\langle \mathbf{x}_*, i \rangle | \theta_i^{(s)}}^2) - \mu_{\langle \mathbf{x}_*, i \rangle}^2.\end{aligned}$$

The above results can be derived from $\mu_{\langle \mathbf{x}_*, i \rangle} = \mathbb{E}_{q^*(\theta_i)}[\mathbb{E}_{p(f(\mathbf{x}_*) | \theta_i, k_i)}[f(\mathbf{x}_*)]] = \mathbb{E}_{q^*(\theta_i)}[\mu_{\langle \mathbf{x}_*, i \rangle | \theta_i}]$ and $\sigma_{\langle \mathbf{x}_*, i \rangle}^2 = \mathbb{E}_{q^*(\theta_i)}[\sigma_{\langle \mathbf{x}_*, i \rangle | \theta_i}^2] + \mathbb{V}_{q^*(\theta_i)}[\mu_{\langle \mathbf{x}_*, i \rangle | \theta_i}] = \mathbb{E}_{q^*(\theta_i)}[\sigma_{\langle \mathbf{x}_*, i \rangle | \theta_i}^2 + \mu_{\langle \mathbf{x}_*, i \rangle | \theta_i}^2] - \mu_{\langle \mathbf{x}_*, i \rangle}^2$. Then, (15) can be derived in the same manner by marginalizing out k using $q^*(k_i)$.

E Details of Experimental Setup

E.1 Composite Kernels

The composite kernels in \mathcal{K}_{12} (Section 6.1) are shown in Table 1 below:

Table 1: Composite kernel types in \mathcal{K}_{12} .

	Composite Kernel
k1	LIN + RQ
k2	LIN \times RQ + LIN
k3	LIN \times RQ + PER
k4	PER + RQ + SE
k5	PER + LIN + RQ
k6	PER + PER + SE
k7	PER \times SE + SE
k8	PER \times RQ + SE
k9	PER \times LIN + SE
k10	PER \times LIN \times SE
k11	PER \times LIN \times RQ
k12	(PER + RQ) \times LIN

The following true kernels are used to generate the synthetic datasets. The three base kernels in (PER + RQ) \times LIN are

$$\begin{aligned}k_{\text{PER}}(\mathbf{x}, \mathbf{x}') &\triangleq 0.1^2 \exp\left(-\frac{2 \sin^2(\pi |\mathbf{x} - \mathbf{x}'| / 2\pi)}{2^2}\right) \\ k_{\text{RQ}}(\mathbf{x}, \mathbf{x}') &\triangleq 0.1^2 \left(1 + \frac{(\mathbf{x} - \mathbf{x}')^2}{2 \times 1 \times 3^2}\right)^{-1} \\ k_{\text{LIN}}(\mathbf{x}, \mathbf{x}') &\triangleq \frac{\mathbf{x}\mathbf{x}'}{5^2}.\end{aligned}$$

The three base kernels in PER \times LIN \times RQ are

$$\begin{aligned}k_{\text{PER}}(\mathbf{x}, \mathbf{x}') &\triangleq 0.1^2 \exp\left(-\frac{2 \sin^2(\pi |\mathbf{x} - \mathbf{x}'| / 2\pi)}{1^2}\right) \\ k_{\text{RQ}}(\mathbf{x}, \mathbf{x}') &\triangleq 0.1^2 \left(1 + \frac{(\mathbf{x} - \mathbf{x}')^2}{2 \times 1 \times 8^2}\right)^{-1} \\ k_{\text{LIN}}(\mathbf{x}, \mathbf{x}') &\triangleq \frac{\mathbf{x}\mathbf{x}'}{3^2}.\end{aligned}$$

E.2 Additional Experimental Results

Tables 2 and 3 show the final RMSEs incurred by VBKS and VBKS-s for 10 independent runs that correspond to the results in Figs. 3b and 4b (i.e., showing averaged RMSEs with standard deviation), respectively. For both datasets, it can be observed that VBKS consistently incurs smaller RMSEs than VBKS-s in all 10 independent runs.

Table 2: RMSEs (MWh) for 10 runs for Swissgrid dataset.

run	VBKS-s	VBKS
1	160.6	155.9
2	162.1	159.3
3	162.0	159.9
4	159.4	156.2
5	159.0	156.2
6	159.2	156.7
7	159.2	156.9
8	163.3	158.4
9	158.4	155.0
10	158.6	155.0

Table 3: RMSEs ($^{\circ}\text{C}$) for 10 runs for IEQ dataset.

run	VBKS-s	VBKS
1	0.416	0.380
2	0.380	0.371
3	0.383	0.373
4	0.386	0.374
5	0.379	0.367
6	0.386	0.372
7	0.380	0.370
8	0.384	0.371
9	0.395	0.378
10	0.390	0.377