Abstract

Large-scale Gaussian process inference has long faced practical challenges due to time and space complexity that is superlinear in dataset size. While sparse variational Gaussian process models are capable of learning from large-scale data, standard strategies for sparsifying the model can prevent the approximation of complex functions. In this work, we propose a novel variational Gaussian process model that decouples the representation of mean and covariance functions in reproducing kernel Hilbert space. We show that this new parametrization generalizes previous models. Furthermore, it yields a variational inference problem that can be solved by stochastic gradient ascent with time and space complexity that is only linear in the number of mean function parameters, regardless of the choice of kernels, likelihoods, and inducing points. This strategy makes the adoption of large-scale expressive Gaussian process models possible. We run several experiments on regression tasks and show that this decoupled approach greatly outperforms previous sparse variational Gaussian process inference procedures.

1 Introduction

Gaussian process (GP) inference is a popular nonparametric framework for reasoning about functions under uncertainty. However, the expressiveness of GPs comes at a price: solving (approximate) inference for a GP with \( N \) data instances has time and space complexities in \( \Theta(N^3) \) and \( \Theta(N^2) \), respectively. Therefore, GPs have traditionally been viewed as a tool for problems with small- or medium-sized datasets.

Recently, the concept of inducing points has been used to scale GPs to larger datasets. The idea is to summarize a full GP model with statistics on a sparse set of \( M \ll N \) fictitious observations \([18][24]\). By representing a GP with these inducing points, the time and the space complexities are reduced to \( O(NM^2 + M^3) \) and \( O(NM + M^2) \), respectively. To further process datasets that are too large to fit into memory, stochastic approximations have been proposed for regression \([10]\) and classification \([11]\). These methods have similar complexity bounds, but with \( N \) replaced by the size of a mini-batch \( N_m \).

Despite the success of sparse models, the scalability issues of GP inference are far from resolved. The major obstruction is that the cubic complexity in \( M \) in the aforementioned upper-bound is also a lower-bound, which results from the inversion of an \( M \)-by-\( M \) covariance matrix defined on the inducing points. As a consequence, these models can only afford to use a small set of \( M \) basis functions, limiting the expressiveness of GPs for prediction.

In this work, we show that superlinear complexity is not completely necessary. Inspired by the reproducing kernel Hilbert space (RKHS) representation of GPs \([2]\), we propose a generalized variational GP model, called DGP (Decoupled Gaussian Processes), which decouples the bases
While we design SVDGP, we show, when DGPs are used as a variational posterior [24], the associated variational inference can be used. The last two columns list the time and space complexity. Each algorithm can be viewed as a special way to solve the maximization problem can be solved by stochastic gradient ascent with space complexity $O(M^3)$, which allows us to keep the time and space complexity similar to previous methods (by choosing $M_\beta = M$) while greatly increasing accuracy. To the best of our knowledge, this is the first variational GP algorithm that admits linear complexity.

<table>
<thead>
<tr>
<th></th>
<th>Time</th>
<th>Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVI [10]</td>
<td>$O(DN_\alpha N M_\alpha^2 + N M_\alpha M_\beta^2)$</td>
<td>$O(NM_\alpha + M_\alpha^2)$</td>
</tr>
<tr>
<td>VSGPR [2]</td>
<td>$O(DNM + NM^2 + M^3)$</td>
<td>$O(NM + M^2)$</td>
</tr>
<tr>
<td>SVDGP</td>
<td>$O(DN^2 + N^3)$</td>
<td>$O(N^2)$</td>
</tr>
</tbody>
</table>

Table 1: Comparison between SVDGP and variational GPR algorithms: SVI [10], iVSGPR [2], VSGPR [24], and GPR [19], where $N$ is the number of observations/the size of a mini-batch, $M$, $M_\alpha$, $M_\beta$ are the number of basis functions, and $D$ is the input dimension. Here it is assumed $M_\alpha \geq M_\beta$.

![Figure 1](image1.png)

Figure 1: Comparison between models with shared and decoupled basis. (a)(c) denote the models with shared basis of size $M$. (b) denotes the model of decoupled basis with size $(M_\alpha, M_\beta)$. In each figure, the red line denotes the ground truth; the blue circles denote the observations; the black line and the gray area denote the mean and variance in prediction, respectively.

for the mean and the covariance functions. Specifically, let $M_\alpha$ and $M_\beta$ be the numbers of basis functions used to model the mean and the covariance functions, respectively. Assume $M_\alpha \geq M_\beta$. We show, when DGPs are used as a variational posterior [24], the associated variational inference problem can be solved by stochastic gradient ascent with space complexity $O(N M_\alpha + M_\alpha^2)$ and time complexity $O(DN_\alpha N M_\alpha + N M_\alpha M_\beta + M_\beta^2)$, where $D$ is the input dimension. We name this algorithm SVDGP. As a result, we can choose $M_\alpha \gg M_\beta$, which allows us to keep the time and space complexity similar to previous methods (by choosing $M_\beta = M$) while greatly increasing accuracy. To the best of our knowledge, this is the first variational GP algorithm that admits linear complexity in $M_\alpha$, without any assumption on the choice of kernel and likelihood.

While we design SVDGP for general likelihoods, in this paper we study its effectiveness in Gaussian process regression (GPR) tasks. We consider this is without loss of generality, as most of the sparse variational GPR algorithms in the literature can be modified to handle general likelihoods by introducing additional approximations (e.g. in Hensman et al. [11] and Sheth et al. [22]). Our experimental results show that SVDGP significantly outperforms the existing techniques, achieving higher variational lower bounds and lower prediction errors when evaluated on held-out test sets.

1.1 Related Work

Our framework is based on the variational inference problem proposed by Titsias [24], which treats the inducing points as variational parameters to allow direct approximation of the true posterior. This is in contrast to Seeger et al. [21], Snelson and Ghahramani [23]. Quiñonero-Candela and Rasmussen [18], and Lázaro-Gredilla et al. [15], which all use inducing points as hyper-parameters of a degenerate prior. While both approaches have the same time and space complexity, the latter additionally introduces a large set of unregularized hyper-parameters and, therefore, is more likely to suffer from over-fitting.

In Table 1 we compare SVDGP with recent GPR algorithms in terms of the assumptions made and the time and space complexity. Each algorithm can be viewed as a special way to solve the maximization of the variational lower bound [5], presented in Section 3.2. Our algorithm SVDGP generalizes the previous approaches to allow the basis functions for the mean and the covariance to be decoupled, so an approximate solution can be found by stochastic gradient ascent in linear complexity.

---

1The first three columns show the algorithms to update the parameters: SGA/SNGA/SMA denotes stochastic gradient/natural gradient/mirror ascent, and CG denotes batch nonlinear conjugate gradient ascent. The 4th and the 5th columns indicate whether the bases for mean and covariance are strictly shared, and whether a variational posterior can be used. The last two columns list the time and space complexity.
To illustrate the idea, we consider a toy GPR example in Figure 1. The dataset contains 500 noisy observations of a sinc function. Given the same training data, we conduct experiments with three different GP models. Figure 1(a)(c) show the results of the traditional coupled basis, which can be solved by any of the variational algorithms listed in Table 1 and Figure 1(b) shows the result using the decoupled approach SVDGP. The sizes of basis and observations are selected to emulate a large dataset scenario. We can observe SVDGP achieves a nice trade-off between prediction performance and complexity: it achieves almost the same accuracy in prediction as the full-scale model in Figure 1(c) and preserves the overall shape of the predictive variance.

In addition to the sparse algorithms above, some recent attempts aim to revive the non-parametric property of GPs by structured covariance functions. For example, Wilson and Nickisch [27] proposes to space the inducing points on a multidimensional lattice, so the time and space complexities of using a product kernel becomes $O(N + DM^{1+1/L})$ and $O(N + DM^{1+2/L})$, respectively. However, because $M = c^D$, where c is the number of grid points per dimension, the overall complexity is exponential in $D$ and infeasible for high-dimensional data. Another interesting approach by Hensman et al. [12] combines variational inference [24] and a sparse spectral approximation [15]. By equally spacing inducing points on the spectrum, they show the covariance matrix on the inducing points have diagonal plus low-rank structure. With MCMC, the algorithm can achieve complexity $O(DNM)$. However, the proposed structure in [12] does not help to reduce the complexity when an approximate Gaussian posterior is favored or when the kernel hyper-parameters need to be updated.

Other kernel methods with linear complexity have been proposed using functional gradient descent [14, 15]. However, because these methods use a model strictly the same size as the entire dataset, they fail to estimate the predictive covariance, which requires $\Omega(N^2)$ space complexity. Moreover, they cannot learn hyper-parameters online. The latter drawback also applies to greedy algorithms based on rank-one updates, e.g. the algorithm of Csató and Opper [4].

In contrast to these previous methods, our algorithm applies to all choices of inducing points, likelihoods, and kernels, and we allow both variational parameters and hyper-parameters to adapt online as more data are encountered.

2 Preliminaries

In this section, we briefly review the inference for GPs and the variational framework proposed by Titsias [24]. For now, we will focus on GPR for simplicity of exposition. We will discuss the case of general likelihoods in the next section when we introduce our framework, DGP.

2.1 Inference for GPs

Let \( f : \mathcal{X} \to \mathbb{R} \) be a latent function defined on a compact domain \( \mathcal{X} \subset \mathbb{R}^D \). Here we assume \textit{a priori} that \( f \) is distributed according to a Gaussian process \( \mathcal{GP}(m, k) \). That is, \( \forall x, x' \in \mathcal{X}, \mathbb{E}[f(x)] = m(x) \) and \( \mathbb{C}[f(x), f(x')] = k(x, x') \). In short, we write \( f \sim \mathcal{GP}(m, k) \).

A GP probabilistic model is composed of a likelihood \( p(y|f(x)) \) and a GP prior \( \mathcal{GP}(m, k) \); in GPR, the likelihood is assumed to be Gaussian i.e. \( p(y|f(x)) = \mathcal{N}(y|f(x), \sigma^2) \) with variance \( \sigma^2 \). Usually, the likelihood and the GP prior are parameterized by some hyper-parameters, which we summarize as \( \theta \). This includes, for example, the variance \( \sigma^2 \) and the parameters implicitly involved in defining \( k(x, x') \). For notational convenience, and without loss of generality, we assume \( m(x) = 0 \) in the prior distribution and omit explicitly writing the dependence of distributions on \( \theta \).

Assume we are given a dataset \( D = \{(x_n, y_n)\}_{n=1}^N \), in which \( x_n \in \mathcal{X} \) and \( y_n \sim p(y|f(x_n)) \). Let \( X = \{x_n\}_{n=1}^N \) and \( y = \{y_n\}_{n=1}^N \). Inference for GPs involves solving for the posterior \( p_{\theta^*}(f(x)|y) \) for any new input \( x \in \mathcal{X} \), where \( \theta^* = \arg\max_{\theta} \log p_{\theta}(y) \). For example in GPR, because the likelihood is Gaussian, the predictive posterior is also Gaussian with mean and covariance

\[
m_{y|x}(x) = k_{x,X}(K_{X} + \sigma^2 I)^{-1}y, \quad k_{y|x}(x, x') = k_{x,x'} - k_{x,X}(K_{X} + \sigma^2 I)^{-1}k_{X,x'},
\]

and the hyper-parameter \( \theta^* \) can be found by nonlinear conjugate gradient ascent [19]

\[
\max_{\theta} \log p_{\theta}(y) = \max_{\theta} \log \mathcal{N}(y|0, K_{X} + \sigma^2 I),
\]

In notation, we use boldface to distinguish finite-dimensional vectors (lower-case) and matrices (upper-case) that are used in computation from scalar and abstract mathematical objects.
where \( k_{.,\cdot}, k_{.,} \), \( \textbf{K}_{,} \), denote the covariances between the sets in the subscript. One can show that these two functions, \( m_y(x) \) and \( k_y(x,x') \), define a valid GP. Therefore, given observations \( y \), we say \( f \sim \text{GP}(m_y, k_y) \).

Although theoretically GPs are non-parametric and can model any function as \( N \to \infty \), in practice it is difficult. As the inference has time complexity \( \Omega(N^3) \) and space complexity \( \Omega(N^2) \), applying vanilla GPs to large datasets is infeasible.

2.2 Variational Inference with Sparse GPs

To scale GPs to large datasets, Titsias [24] introduced a scheme to compactly approximate the true posterior with a sparse GP, \( \text{GP}(\hat{m}_y, \hat{k}_y) \), defined by the statistics on \( M \ll N \) function values: \( \{L_m f(x_m)\}^M_{m=1} \), where \( L_m \) is a bounded linear operator and \( x_m \in \mathcal{X} \). \( L_m f(\cdot) \) is called an inducing function and \( x_m \) an inducing point. Common choices of \( L_m \) include the identity map (as used originally by Titsias [24]) and integrals to achieve better approximation or to consider multi-domain information [26, 7, 3]. Intuitively, we can think of \( \{L_m f(x_m)\}^M_{m=1} \) as a set of potentially indirect observations that capture salient information about the unknown function \( f \).

Titsias [24] solves for \( \text{GP}(\hat{m}_y, \hat{k}_y) \) by variational inference. Let \( \tilde{X} = \{x_m\}^M_{m=1} \) and let \( f_X \in \mathbb{R}^N \) and \( f_{\tilde{X}} \in \mathbb{R}^M \) be the (inducing) function values defined on \( X \) and \( \tilde{X} \), respectively. Let \( p(f_{\tilde{X}}) \) be the prior given by \( \text{GP}(m, k) \) and define \( q(f_{\tilde{X}}) = \mathcal{N}(f_{\tilde{X}} | \hat{m}, \hat{\textbf{S}}) \) to be its variational posterior, where \( \hat{m} \in \mathbb{R}^M \) and \( \hat{\textbf{S}} \in \mathbb{R}^{M \times M} \) are the mean and the covariance of the approximate posterior of \( f_{\tilde{X}} \). Titsias [24] proposes to use \( q(f_{X}, f_{\tilde{X}}) = p(f_X | f_{\tilde{X}}) q(f_{\tilde{X}}) \) as the variational posterior to approximate \( p(f_X, f_{\tilde{X}} | y) \) and to solve for \( q(f_{\tilde{X}}) \) together with the hyper-parameter \( \theta \) through

\[
\max_{\theta, \tilde{X}, \hat{m}, \hat{\textbf{S}}} L_{\theta}(\tilde{X}, \hat{m}, \hat{\textbf{S}}) = \max_{\theta, \tilde{X}, \hat{m}, \hat{\textbf{S}}} \int q(f_X, f_{\tilde{X}}) \log \frac{p(y | f_X) p(f_X | f_{\tilde{X}}) p(f_{\tilde{X}})}{q(f_X, f_{\tilde{X}})} df_X df_{\tilde{X}}, \tag{3}
\]

where \( L_{\theta} \) is a variational lower bound of \( \log p_0(y) \), \( p(f_X | f_{\tilde{X}}) = \mathcal{N}(f_X | K_{X,\tilde{X}} K_{\tilde{X}}^{-1} f_{\tilde{X}}, K_X - K_{\tilde{X}} \) is the conditional probability given in \( \text{GP}(m, k) \), and \( K_X = K_{X,\tilde{X}} K_{\tilde{X}}^{-1} K_{\tilde{X},X} \).

At first glance, the specific choice of variational posterior \( q(f_X, f_{\tilde{X}}) \) seems heuristic. However, although parameterized finitely, it resembles a full-fledged GP \( \text{GP}(\hat{m}_y, \hat{k}_y) \):

\[
\hat{m}_y(x) = k_{x,\cdot} K_{\cdot,\cdot}^{-1} \hat{m}, \quad \hat{k}_y(x, x') = k_{x,x'} + k_{x,\cdot} K_{\cdot,\cdot}^{-1} (\hat{\textbf{S}} - K_{\cdot,\cdot}) K_{\cdot,\cdot}^{-1} k_{\cdot,x'}. \tag{4}
\]

This result is further studied in Matthews et al. [10] and Cheng and Boots [2], where it is shown that \((3)\) is indeed minimizing a proper KL-divergence between Gaussian processes/measure.

By comparing \((2)\) and \((3)\), one can show that the time and the space complexities now reduce to \( O(D N M + M^2 N + M^3) \) and \( O(M^2 + MN) \), respectively, due to the low-rank structure of \( \textbf{K}_{X} \) [24]. To further reduce complexity, stochastic optimization, such as stochastic natural ascent [10] or stochastic mirror descent [2] can be applied. In this case, \( N \) in the above asymptotic bounds would be replaced by the size of a mini-batch \( N_m \). The above results can be modified to consider general likelihoods as in [22, 11].

3 Variational Inference with Decoupled Gaussian Processes

Despite the success of sparse GPs, the scalability issues of GPs persist. Although parameterizing a GP with inducing points/functions enables learning from large datasets, it also restricts the expressiveness of the model. As the time and the space complexities still scale in \( \Omega(M^3) \) and \( \Omega(M^2) \), we cannot learn or use a complex model with large \( M \).

In this work, we show that these two complexity bounds, which have long accompanied GP models, are not strictly necessary, but are due to the tangled representation canonically used in the GP.
literature. To elucidate this, we adopt the dual representation of Cheng and Boots [2], which treats

GPs as linear operators in RKHS. But, unlike Cheng and Boots [2], we show how to decouple the

basis representation of mean and covariance functions of a GP and derive a new variational problem,

which can be viewed as a generalization of (3). We show that this problem—with arbitrary likelihoods

and kernels—can be solved by stochastic gradient ascent with linear complexity in Ms, the number

of parameters used to specify the mean function for prediction.

In the following, we first review the results in [2]. We next introduce the decoupled representation,

DGP, and its variational inference problem. Finally, we present SVDGP and discuss the case with

general likelihoods.

3.1 Gaussian Processes as Gaussian Measures

Let an RKHS \( \mathcal{H} \) be a Hilbert space of functions with the reproducing property: \( \forall x \in \mathcal{X}, \exists \phi_x \in \mathcal{H} \) such that \( \forall f \in \mathcal{H}, f(x) = \phi_x^T f \) A Gaussian process \( \mathcal{GP}(m,k) \) is equivalent to a Gaussian measure \( \nu \) on Banach space \( \mathcal{B} \) which possesses an RKHS \( \mathcal{H} \) such that for any \( x, x' \in \mathcal{X} \), \( \exists \phi_x, \phi_{x'} \in \mathcal{H} \), we can write \( m(x) = \phi_x^T \mu \) and \( k(x,x') = \phi_x^T \Sigma \phi_{x'} \). The triple \( (\mathcal{B}, \nu, \mathcal{H}) \) is known as an abstract Wiener space [3][4], in which \( \mathcal{H} \) is also called the Cameron-Martin space. Here the restriction that \( \mu, \Sigma \) are RKHS objects is necessary, so the variational inference problem in the next section can be well-defined.

We call this the dual representation of a GP in RKHS \( \mathcal{H} \) (the mean function \( m \) and the covariance function \( k \) are realized as linear operators \( \mu \) and \( \Sigma \) defined in \( \mathcal{H} \)). With abuse of notation, we write \( \mathcal{N}(f|\mu, \Sigma) \) in short. This notation does not mean a GP has a Gaussian distribution in \( \mathcal{H} \), nor does it imply that the sample paths from \( \mathcal{GP}(m,k) \) are necessarily in \( \mathcal{H} \). Precisely, \( \mathcal{B} \) contains the sample paths of \( \mathcal{GP}(m,k) \) and \( \mathcal{H} \) is dense in \( \mathcal{B} \). In most applications of GP models, \( \mathcal{B} \) is the Banach space of continuous function \( C(\mathcal{X}, \mathcal{Y}) \) and \( \mathcal{H} \) is the span of the covariance function. As a special case, if \( \mathcal{H} \) is finite-dimensional, \( \mathcal{B} \) and \( \mathcal{H} \) coincide and \( \nu \) becomes equivalent to a Gaussian distribution in a Euclidean space.

In relation to our previous notation in Section 2.1, suppose \( k(x, x') = \phi_x^T \phi_{x'} \) and \( \phi_x : \mathcal{X} \rightarrow \mathcal{H} \) is a feature map to some Hilbert space \( \mathcal{H} \). Then we have assumed a priori that \( \mathcal{GP}(m,k) = \mathcal{N}(f|0, I) \) is a normal Gaussian measure; that is \( \mathcal{GP}(m,k) \) samples functions \( f \) in the form \( f(x) = \sum_{i=1}^{\dim \mathcal{H}} \phi_i(x)^T \epsilon_i, \) where \( \epsilon_i \sim \mathcal{N}(0, 1) \) are independent. Note if \( \dim \mathcal{H} = \infty \), with probability one \( f \) is not in \( \mathcal{H} \), but fortunately \( \mathcal{H} \) is large enough for us to approximate the sampled functions. In particular, it can be shown that the posterior \( \mathcal{GP}(m|y, k|y) \) in GPR has a dual RKHS representation in the same RKHS as the prior GP [2].

3.2 Variational Inference in Gaussian Measures

Cheng and Boots [2] proposes a dual formulation of (3) in terms of Gaussian measures:

\[
\max_{q(f), \theta} \mathcal{L}_\theta(q(f)) = \max_{q(f), \theta} \frac{1}{2} \int q(f) \log \frac{p(y|f)p(f)}{q(f)} df - \frac{1}{2} \max_{q(f), \theta} \mathbb{E}_q[\log p_0(y|f)] - \text{KL}[q(||p||_2),
\]

(5)

where \( q(f) = \mathcal{N}(f|\mu, \Sigma) \) is a variational Gaussian measure and \( p(f) = \mathcal{N}(f|0, I) \) is a normal prior. Its connection to the inducing points/functions in [3] can be summarized as follows [2][3]: Define a linear operator \( \Psi_X : \mathbb{R}^M \rightarrow \mathcal{H} \) as \( a \mapsto \sum_{m=1}^{M} a_m \psi_m \), where \( \psi_m \in \mathcal{H} \) is defined such that \( \psi_m^T \mu = \mathbb{E}[L_m f(x_m)] \). Then (3) and (5) are equivalent, if \( q(f) \) has a subspace parametrization,

\[
\hat{\mu} = \Psi_X a, \quad \hat{\Sigma} = I + \Psi_X A \Psi_X^T
\]

(6)

with \( a \in \mathbb{R}^M \) and \( A \in \mathbb{R}^{M \times M} \) satisfying \( \hat{\mu} = K_X a \), and \( \hat{\Sigma} = K_X + K_X A K_X^T \). In other words, the variational inference algorithms in the literature are all using a variational Gaussian measure in which \( \hat{\mu} \) and \( \hat{\Sigma} \) are parametrized by the same basis \( \{\psi_m| x_m \in X\}_{i=1}^{M} \).

3 To simplify the notation, we write \( \phi_x^T f \) for \( \langle f, \phi_x \rangle_\mathcal{H} \), and \( f^T L g \) for \( \langle f, L g \rangle_\mathcal{H} \), where \( f, g \in \mathcal{H} \) and \( L : \mathcal{H} \rightarrow \mathcal{H} \), even if \( \mathcal{H} \) is infinite-dimensional.

4 Such \( \mathcal{H} \) w.l.o.g. can be identified as the natural RKHS of the covariance function of a zero-mean prior GP.

5 We assume \( q(f) \) is absolutely continuous wrt \( p(f) \), which is true as \( p(f) \) is non-degenerate. The integral denotes the expectation of \( \log p_0(q(f)) + \log p(q(f)) \) over \( q(f) \), and \( \frac{\partial f}{\partial x} \) denotes the Radon-Nikodym derivative.
We notice that while the structure of (8) looks similar to (4), directly replacing the basis when we select \(M\) whereas we parametrize directly by (6) with \(\Psi\) for scalability and \(H\) for better stability (which always reduces the uncertainty in the posterior compared with the prior). The decoupled subspace parametrization (7) corresponds to a DGP, \(\text{DGP}(\hat{m}_{\tilde{x}y}^\alpha, \hat{k}_{\tilde{x}y}^\beta)\), with mean and covariance functions as:

\[
\hat{m}_{\tilde{x}y}^\alpha(x) = k_{x,\alpha} \Psi, \quad \hat{k}_{\tilde{x}y}^\alpha(x, x') = k_{x,\beta} (B^{-1} + K_\beta)^{-1} k_{x,\alpha}. \tag{8}
\]

3.3 Disentangling the GP Representation with DGPs

While Cheng and Boots \[2\] treat (5) as an equivalent form of (3), here we show that it is a generalization. By further inspecting (5), it is apparent that sharing the basis \(\Psi\) between \(\tilde{\mu}\) and \(\tilde{\Sigma}\) in (6) is not strictly necessary, since (5) seeks to optimize two linear operators, \(\mu\) and \(\Sigma\). With this in mind, we propose a new parametrization that decouples the bases for \(\mu\) and \(\Sigma\):

\[
\hat{\mu} = \Psi_{\alpha} a, \quad \hat{\Sigma} = (I + \Psi_{\beta} B \Psi_{\beta}^T)^{-1} \tag{7}
\]

where \(\Psi_{\alpha} : \mathbb{R}^{M_\alpha} \to \mathcal{H}\) and \(\Psi_{\beta} : \mathbb{R}^{M_\beta} \to \mathcal{H}\) denote linear operators defined similarly to \(\Psi_{\tilde{\mu}}\) and \(\Psi_{\tilde{\Sigma}}\) through its inversion with \(B\) so the condition that \(\hat{\Sigma} \succeq 0\) can be easily realized as \(B \succeq 0\). This form agrees with the posterior covariance in GPR \[2\] and will give a posterior that is strictly less uncertain than the prior.

The decoupled subspace parametrization (7) corresponds to a DGP, \(\mathcal{GP}(\hat{m}_{\tilde{x}y}^\alpha, \hat{k}_{\tilde{x}y}^\beta)\), with mean and covariance functions as:

\[
\hat{m}_{\tilde{x}y}^\alpha(x) = k_{x,\alpha} \Psi, \quad \hat{k}_{\tilde{x}y}^\alpha(x, x') = k_{x,\beta} (B^{-1} + K_\beta)^{-1} k_{x,\alpha}. \tag{8}
\]

3.4 SVDGP: Algorithm and Analysis

Substituting the decoupled subspace parametrization (7) into the variational inference problem in (5) results in a numerical optimization problem: \(\max_{q(f, \theta)} \mathbb{E}_q[\log p_\theta(y|f)] - \mathbb{KL}[q||p]\) with

\[
\begin{align*}
\mathbb{KL}[q||p] &= \frac{1}{2} a^T K_\alpha a + \frac{1}{2} \log |I + K_\beta B| + \frac{-1}{2} \text{tr} (K_\beta (B^{-1} + K_\beta)^{-1}) \\
\mathbb{E}_q[\log p_\theta(y|f)] &= \sum_{n=1}^N \mathbb{E}_{q(f(x_n))}[\log p_\theta(y_n|f(x_n))] \tag{9}
\end{align*}
\]

where each expectation is over a scalar Gaussian \(q(f(x_n))\) given by (8) as functions of \((a, \alpha)\) and \((B, \beta)\). Our objective function contains (11) as a special case, which assumes \(\alpha = \beta = \tilde{X}\). In addition, we note that Hensman et al. (11) indirectly parametrize the posterior by \(\hat{m}\) and \(S = LL^T\), whereas we parametrize directly by \(\Theta\) with a for scalability and \(B = LL^T\) for better stability (which always reduces the uncertainty in the posterior compared with the prior).

We notice that \((a, \alpha)\) and \((B, \beta)\) are completely decoupled in (9) and potentially combined again in (10). In particular, if \(p_\theta(y_n|f(x_n))\) is Gaussian as in GPR, we have an additional decoupling, i.e. \(\tilde{L}_\theta(a, B, \alpha, \beta) = \mathcal{F}_\theta(a, \alpha) + \mathcal{G}_\theta(B, \beta)\) for some \(\mathcal{F}_\theta(a, \alpha)\) and \(\mathcal{G}_\theta(B, \beta)\). Intuitively, the optimization over \((a, \alpha)\) aims to minimize the fitting-error, and the optimization over \((B, \beta)\) aims to memorize the samples encountered so far; the mean and the covariance functions only interact indirectly through the optimization of the hyper-parameter \(\theta\).

One salient feature of SVDGP is that it tends to overestimate, rather than underestimate, the variance, when we select \(M_\beta \leq M_\alpha\). This is inherited from the non-degeneracy property of the variational

\[\text{In practice, we can parametrize } B = LL^T \text{ with Cholesky factor } L \in \mathbb{R}^{M_\beta \times M_\beta} \text{ so the problem is unconstrained. The required terms in (8) and later in (9) can be stably computed as } \Psi (B^{-1} + K_\beta)^{-1} = LH^{-1}L^T \text{ and } \log |I + K_\beta B| = \log |H|, \text{ where } H = I + L^T K_\beta L.\]
We compare our new algorithm, Θ, and see the covariance in (8) becomes the same as the prior; moreover, the objective function can be computed in O(\|B\|^2) or O(\|B\|). By (9), One can show that the complexity to compute \(\nabla_a KL[q||p] = K_a a\) and \(\nabla_B KL[q||p] = \frac{1}{2}(I+K_a B)^{-1}K_a B K_B(I+K_B)^{-1}\). Therefore, the time complexity to compute \(\nabla_a KL[q||p]\) can be reduced to O(\(N_m M_a\)) if we sample over the columns of \(K_a\) with a mini-batch of size \(N_m\). By contrast, the time complexity to compute \(\nabla_B KL[q||p]\) will always be O(\(M_B^3\)) and cannot be further reduced, regardless of the parametrization of \(B\).

The gradient with respect to \(\alpha\) and \(\beta\) can be derived similarly and have time complexity O(\(DN_m M_a\)) and O(\(DM_B^2 + M_B^3\)), respectively.

**Expected Log-Likelihood** Let \(\bar{m}(a, \alpha) \in \mathbb{R}^N\) and \(\bar{s}(B, \beta) \in \mathbb{R}^N\) be the vectors of the mean and covariance of scalar Gaussian \(q(f(x_n))\) for \(n \in \{1, \ldots, N\}\). As (10) is a sum over \(N\) terms, by sampling with a mini-batch of size \(N_m\), an unbiased gradient of (10) with respect to \((\theta, \bar{m}, \bar{s})\) can be computed in O(\(N_m\)). To compute the full gradient with respect to \((a, B, \alpha, \beta)\), we compute the derivative of \(\bar{m}\) and \(\bar{s}\) with respect to \((a, B, \alpha, \beta)\) and then apply chain rule. These steps take O(\(DN_m M_a\)) and O(\(DN_m M_B + N_m M_B^2 + M_B^3\)) for \((a, \alpha)\) and \((B, \beta)\), respectively.

The above analysis shows that the curse of dimensionality in GPs originates in the covariance function. For space complexity, the decoupled parametrization (7) requires memory in O(\(N_m M_a + M_B^3\)); for time complexity, an unbiased gradient with respect to \((a, \alpha)\) can be computed in O(\(DN_m M_a\)), but that with respect to \((B, \beta)\) has time complexity O(\(DN_m M_B + N_m M_B^2 + M_B^3\)). This motivates choosing \(M_B = O(M)\) and \(M_a\) in O(\(M_B^3\)) or O(\(M_B^3\)), which maintains the same complexity as previous variational techniques but greatly improves the prediction performance.

### 4 Experimental Results

We compare our new algorithm, SVDGP, with the state-of-the-art incremental algorithms for sparse variational GPR, SVI [10] and iViG [2], as well as the classical GPR and the batch algorithm VS-GPR [24]. As discussed in Section 4.3, these methods can be viewed as different ways to optimize (5).

Therefore, in addition to the normalized mean square error (nMSE) [19] in prediction, we report the performance in the variational lower bound (VLB) [5], which also captures the quality of the predictive variance and hyper-parameter learning [10]. These two metrics are evaluated on held-out test sets in all of our experimental domains.

---

[1] Due to \(K_B\), the complexity would remain as O(\(M_B^3\)) even if \(B\) is constrained to be diagonal.

[2] The exact marginal likelihood is computationally infeasible to evaluate for our large model.

---

**Algorithm 1** Online Learning with DGPs

**Parameters:** \(M_s, M_B, N_m, N_D\)

**Input:** \(\mathcal{M}(a, B, \alpha, \beta, \theta), D\)

1: \(\theta_0 \leftarrow \text{initializedHyperparameters}(\text{sampleMinibatch}(D, N_m))\)
2: for \(t = 1 \ldots T\) do
3: \(D_t \leftarrow \text{sampleMinibatch}(D, N_m)\)
4: \(\mathcal{M}.\text{addBasis}(D_t, N_D, M_s, M_B)\)
5: \(\mathcal{M}.\text{updateModel}(D_t, t)\)
6: end for
We observe that SVDGP while each output is treated independently during learning, Table 2 presents the mean and the standard deviation over all the outputs as the selected metrics are normalized. For the complete experimental results, please refer to Appendix C.

Appendix B for details. We note that all algorithms in comparison use the same kernel and optimize the generalized SE-ARD kernel \([2]\) as the inducing functions in the variational posterior (see Appendix B for details). We note that all algorithms in comparison use the same kernel and optimize both the variational parameters (including inducing points) and the hyperparameters.

In particular, we implement SGA by ADAM \([13]\) (with default parameters \(\beta_1 = 0.9\) and \(\beta_2 = 0.999\)). The step-size for each stochastic algorithm is scheduled according to \(\gamma_t = \gamma_0 (1 + 0.1 \sqrt{T})^{-1},\) where \(\gamma_0 \in \{10^{-1}, 10^{-2}, 10^{-3}\}\) is selected manually for each algorithm to maximize the improvement in objective function after the first 100 iterations. We test each stochastic algorithm for \(T = 2000\) iterations with mini-batches of size \(N_m = 1024\) and the increment size \(N_A = 128\). Finally, the model sizes used in the experiments are listed as follows: \(M_a = 128^2\) and \(M_b = 128\) for SVDGP; \(M = 1024\) for SVI; \(M = 256\) for iSVGPR; \(M = 1024, N = 4096\) for VSGPR; \(N = 1024\) for GP. These settings share similar order of time complexity in our current Matlab implementation.

4.1 Datasets

**Inverse Dynamics of KUKA Robotic Arm** This dataset records the inverse dynamics of a KUKA arm performing rhythmic motions at various speeds \([17]\). The original dataset consists of two parts: KUKA1 and KUKA2, each of which have 17,560 offline data and 180,360 online data with 28 attributes and 7 outputs. In the experiment, we mix the online and the offline data and then split 90% as training data (178,128 instances) and 10% testing data (19,792 instances) to satisfy the i.i.d. assumption.

**Walking MuJoCo** MuJoCo (Multi-Joint dynamics with Contact) is a physics engine for research in robotics, graphics, and animation, created by \([25]\). In this experiment, we gather 1,000 walking trajectories by running TRPO \([20]\). In each time frame, the MuJoCo transition dynamics have a 23-dimensional input and a 17-dimensional output. We consider two regression problems to predict 9 of the 17 outputs from the input \([2]\), MUJOCO1 which maps the input of the current frame (23 dimensions) to the output, and MUJOCO2 which maps the inputs of the current and the previous frames (46 dimensions) to the output. In each problem, we randomly select 90% of the data as training data (842,745 instances) and 10% as test data (93,608 instances).

4.2 Results

We summarize part of the experimental results in Table 2 in terms of nMSE in prediction and VLB. While each output is treated independently during learning, Table 2 presents the mean and the standard deviation over all the outputs as the selected metrics are normalized. For the complete experimental results, please refer to Appendix C.

We observe that SVDGP consistently outperforms the other approaches with much higher VLBs and much lower prediction errors; SVDGP also has smaller standard deviation. These results validate our assumptions.

---

\(^{11}\) The algorithms differ only in whether the bases are shared and how the model is updated (see Table 1).

\(^{12}\) Because of the structure of MuJoCo dynamics, the rest 8 outputs can be trivially known from the input.
initial hypothesis that adopting a large set of basis functions for the mean can help when modeling complicated functions. iVSGPR has the next best result after SVDGP, despite using a basis size of 256, much smaller than that of 1,024 in SVI, VSGPR, and GPR. Similar to SVDGP, iVSGPR also generalizes better than the batch algorithms VSGPR and GPR, which only have access to a smaller set of training data and are more prone to over-fitting. By contrast, the performance of SVI is surprisingly worse than VSGPR. We conjecture this might be due to the fact that the hyper-parameters and the inducing points/functions are only crudely initialized in online learning. We additionally find that the stability of SVI is more sensitive to the choice of step size than other methods. This might explain why in [10, 2] batch data was used to initialize the hyper-parameters and the learning rate to update the hyper-parameters was selected to be much smaller than that for stochastic natural gradient ascent.

To further investigate the properties of different stochastic approximations, we show the change of VLB and the prediction error over iterations and time in Figure 2. Overall, whereas iVSGPR and SVI share similar convergence rate, the behavior of SVDGP is different. We see that iVSGPR converges the fastest, both in time and sample complexity. Afterwards, SVDGP starts to descend faster and surpass the other two methods. From Figure 2 we can also observe that although SVI has similar convergence to iVSGPR, it slows down earlier and therefore achieves a worse result. These phenomenon are observed in multiple experiments.

5 Conclusion

We propose a novel, fully-differentiable framework, Decoupled Gaussian Processes DGP, for large-scale GP problems. By decoupling the representation, we derive a variational inference problem that can be solved with stochastic gradients with linear time and space complexity. Compared with existing algorithms, SVDGP can adopt a much larger set of basis functions to predict more accurately. Empirically, SVDGP significantly outperforms state-of-the-arts variational sparse GPR algorithms in multiple regression tasks. These encouraging experimental results motivate further application of SVDGP to end-to-end learning with neural networks in large-scale, complex real world problems.

Acknowledgments

This work was supported in part by NSF NRI award 1637758.

References


Appendix

A Variational Inference with Decoupled Gaussian Processes

Here we provide the details of the variational inference problem used to learn DGP:

\[
\max_{q(f),\theta} \mathcal{L}_\theta(q(f)) = \max_{q(f),\theta} \int q(f) \log \frac{p_\theta(y|f)p(f)}{q(f)} \,df = \max_{q(f),\theta} E_q[\log p_\theta(y|f)] - KL[q||p], \tag{11}
\]

A.1 KL Divergence

A.1.1 Evaluation

First, we show how to evaluate the KL-divergence. We do so by extending the KL-divergence between two finite-dimensional subspace-parametrized Gaussian measures to infinite dimensional space and show that it is well-defined.

Recall for two \(d\)-dimensional Gaussian distributions \(q(f) = \mathcal{N}(f|\mu, \Sigma)\) and \(p = \mathcal{N}(f|\bar{\mu}, \bar{\Sigma})\), the KL-divergence is given as

Proposition 1.

\[
KL[q||p] := \int \log \frac{q(f)}{p(f)} \,d\mu_q(f) = \int q(f) \log \frac{q(f)}{p(f)} \,df = \frac{1}{2} \left( \text{tr} \left( \Sigma^{-1} \Sigma \right) + (\mu - \bar{\mu})^T \Sigma^{-1} (\mu - \bar{\mu}) + \ln \frac{||\Sigma||}{||\bar{\Sigma}||} - d \right)
\]

Now consider \(q\) and \(p\) are subspace parametrized as

\[
p(f) = \mathcal{N}(f|\bar{\mu}, \bar{\Sigma}) = \mathcal{N}(f|\Psi_\beta a, (I + \Psi_\beta \overline{B} \Psi_\beta^T)^{-1})
\]

\[
q(f) = \mathcal{N}(f|\mu, \Sigma) = \mathcal{N}(f|\Psi_\alpha a, (I + \Psi_\beta \overline{B} \Psi_\beta^T)^{-1}).
\tag{12}
\]

By Proposition 1 we derive the representation of KL-divergence which is applicable even when \(d\) is infinite. Recall in the infinite dimensional case, \(\mu\), \(\Sigma\), \(\bar{\mu}\), and \(\bar{\Sigma}\) are objects in the RKHS \(\mathcal{H}\) (Cameron-Martin space).

Theorem 1. Assume \(q\) and \(p\) are two subspace parametrized Gaussian measures given as (12). Regardless of the dimension of \(\mathcal{H}\), the following holds

\[
KL[q||p] = -\frac{1}{2} \text{tr} \left( (K_\beta + K_\beta \overline{B} K_\beta \overline{B}) (B^{-1} + K_\beta)^{-1} \right) + \frac{1}{2} \log |I + K_\beta B|
\]

\[
+ \frac{1}{2} a^T (K_\alpha + K_\alpha \overline{B} K_\beta \overline{B}) a - a^T (K_{\alpha,\alpha} + K_\alpha \overline{B} K_\beta \overline{B}) a + C
\tag{13}
\]

where

\[
C = \frac{1}{2} \left( \text{tr} (K_\beta B) - \log |I + K_\beta \overline{B}| + a^T (K_\alpha + K_\alpha \overline{B} K_\beta \overline{B}) a \right)
\]

In particular, if \(p\) is normal (i.e. \(p(f) = \mathcal{N}(f|0, I)\)), then

\[
KL[q||p] = \frac{1}{2} a^T K_\alpha a + \frac{1}{2} \log |I + K_\beta B| + \frac{-1}{2} \text{tr} (K_\beta (B^{-1} + K_\beta)^{-1})
\]

Proof.

To prove, we derive each term in (13) as follows.

First, we derive \(\text{tr} (\Sigma^{-1} \Sigma) - d\). Define \(R = (B^{-1} + K_\beta)^{-1}\). Then we can write

\[
\Sigma = (I + \Psi_\beta B \Psi_\beta^T)^{-1} = I - \Psi_\beta (B^{-1} + \Psi_\beta^T \Psi_\beta)^{-1} \Psi_\beta^T = I - \Psi_\beta R \Psi_\beta^T.
\tag{14}
\]

Using (14), we can derive

\[
\Sigma^{-1} \Sigma = (I + \Psi_\beta B \Psi_\beta^T) (I - \Psi_\beta R \Psi_\beta^T) = I + \Psi_\beta B \Psi_\beta^T - \Psi_\beta R \Psi_\beta^T - \Psi_\beta \overline{B} K_\beta \overline{B} R \Psi_\beta^T
\]
We then give the equations to compute the derivatives below. For compactness of notation, we use \( \text{diag} \) to denote element-wise product and use \( 1 \) on automatic differentiation are also applicable and easier to apply. Implementing \( \text{SVDGP} \) purpose here is to show the complexity of calculating the gradients. These equations are useful in implementing \( \text{SVDGP} \).

A.1.2 Gradients

Note this term does not depend on the ambient dimension.

Second, we derive \( \log(|\Sigma|/|\Sigma|) \): Since

\[
\log |\Sigma^{-1}| = \log |B^{-1} + K_\beta B| = \log |I + K_\beta B|.
\]

it holds that

\[
\log \frac{\Sigma}{|\Sigma|} = \log |I + K_\beta B| - \log |I + K_\beta B|.
\]

Finally, we derive the quadratic term:

\[
(\mu - \bar{\mu})^T \Sigma^{-1} (\mu - \bar{\mu})
= \mu^T \Sigma^{-1} \mu - 2\bar{\mu}^T \Sigma^{-1} \mu + \bar{\mu}^T \Sigma^{-1} \bar{\mu}
= a^T \psi_{\alpha}^T \left( I + \psi_{\beta}^T \bar{B} \psi_{\beta}^T \right) \psi_{\alpha} a - 2a^T \psi_{\alpha}^T \left( I + \psi_{\beta}^T \bar{B} \psi_{\beta}^T \right) \psi_{\alpha} \bar{a}
= a^T \left( K_\alpha + K_{\alpha,\beta} B K_{\alpha,\beta} \right) a - 2a^T \left( K_{\alpha,\alpha} + K_{\alpha,\beta} B K_{\alpha,\beta} \right) a + a^T \left( K_\beta + K_{\beta,\beta} B K_{\beta,\beta} \right) a
\]

**Notes to Practitioners**

1. The above expression is well defined even when \( B \geq 0 \), because \( (B^{-1} + K_\beta)^{-1} = B(I + K_\beta B)^{-1} \). Particularly, we can parametrize \( B = L L^T \) with Cholesky factor \( L \in \mathbb{R}^{M_\beta \times M_\beta} \) in practice so the problem is unconstrained. The required terms can be stably computed:

\[
(B^{-1} + K_\beta)^{-1} = L H^{-1} L^T \quad \text{and} \quad \log |I + K_\beta B| = \log |H|, \quad \text{where} \quad H = I + L^T K_\beta L.
\]

2. In implementations, a Jacobi preconditioner can be optionally used. That is, we can parametrize \( a \) and \( L \) through \( a_0 \) and \( L_0 \)

\[
a = \text{diag}(K_\alpha)^{-1} a_0 \quad \text{L} = \text{diag}(K_\beta)^{-1} L_0
\]

for improving stability of using first-order methods, here \( \text{diag} \) denotes the diagonal part.

**A.1.2 Gradients**

Here we derive the equations of the gradient of the variational inference problem of \( \text{SVDGP} \). The purpose here is to show the complexity of calculating the gradients. These equations are useful in implementing \( \text{SVDGP} \) using basic linear algebra routines, while computational-graph libraries based on automatic differentiation are also applicable and easier to apply.

To derive the gradients, we first introduce some short-hand

\[
G_\alpha = K_\alpha + K_{\alpha,\beta} B K_{\beta,\alpha}
G_{\alpha,\alpha} = K_{\alpha,\alpha} + K_{\alpha,\beta} B K_{\beta,\alpha}
G_\beta = K_\beta + K_{\beta,\beta} B K_{\beta,\beta}
\]

and write \( \text{KL}[q||p] \) as

\[
\text{KL}[q||p] = -\frac{1}{2} \text{tr} \left( G_\beta (B^{-1} + K_\beta)^{-1} \right) + \frac{1}{2} \log |I + K_\beta B| + \frac{1}{2} a^T G_\alpha a - a^T G_{\alpha,\alpha} \bar{a}.
\]

We then give the equations to compute the derivatives below. For compactness of notation, we use \( \odot \) to denote element-wise product and use \( 1 \) to denote the vector of ones. In addition, we introduce a linear operator \( \text{diag} \) with overloaded definitions:

1. \( \text{diag} : \mathbb{R}^N \rightarrow \mathbb{R}^{N \times N} \) which constructs a diagonal matrix from a vector
2. \( \text{diag} : \mathbb{R}^{N \times N} \rightarrow \mathbb{R}^N \) which extracts the diagonal elements of a matrix to a vector.

**Proposition 2.** The gradients of \( \text{KL}[q||p] \) is as follows:

\[
\begin{align*}
\nabla_a \text{KL}[q||p] &= G_\alpha a - G_{\alpha, \beta} \bar{a} \\
\nabla_a \text{KL}[q||p] &= \text{diag}(a) (\partial_a G_\alpha a - \partial_a G_{\alpha, \beta} \bar{a}) \\
\nabla_B \text{KL}[q||p] &= \frac{1}{2}(I + K_\beta B)^{-1} (K_\beta BK_\beta - \Delta_\beta) (I + BK_\beta)^{-1} \\
\nabla_\beta \text{KL}[q||p] &= (\partial_\beta K_\beta \circ (B^{-1} + K_\beta)^{-1} G_\beta (B^{-1} + K_\beta)^{-1}) 1 - (\partial_\beta \Delta_\beta \circ (B^{-1} + K_\beta)^{-1}) 1
\end{align*}
\]

where \( \Delta_\beta = G_\beta - K_\beta \) and \( \partial \) is defined as the partial derivative with respect to the left argument. In particular, if the \( p \) is normal,

\[
\begin{align*}
\nabla_a \text{KL}[q||p] &= K_\alpha a \\
\nabla_a \text{KL}[q||p] &= \text{diag}(a) \partial_a K_\alpha a \\
\nabla_B \text{KL}[q||p] &= \frac{1}{2}(I + K_\beta B)^{-1} K_\beta BK_\beta (I + BK_\beta)^{-1} \\
\nabla_\beta \text{KL}[q||p] &= (\partial_\beta K_\beta \circ (B^{-1} + K_\beta)^{-1} K_\beta (B^{-1} + K_\beta)^{-1}) 1
\end{align*}
\]

The derivation of Proposition 2 is only technical, so we omit it here.

Here we only show the derivative with respect to \( B \). Suppose \( B = LL^T \). Then one can apply chain rule and get

\[
\nabla_L \text{KL}[q||p] = 2\nabla_B \text{KL}[q||p] L.
\]

### A.2 Expected Log-Likelihood

#### A.2.1 Evaluation

The evaluation of the expected log-likelihood depends on the mean and covariance in (8), which we repeat here

\[
\hat{m}_y \beta \left(x \right) = k_{x, \alpha} a, \quad \hat{k}_y \beta \left(x, x' \right) = k_{x, \beta} - k_{x, \beta} (B^{-1} + K_\beta)^{-1} k_{\beta, x'}.
\]

Its derivation is trivial by the definition of \( q \) in (12) and (14). For \( N \) observations, the vector form \( \hat{m} \in \mathbb{R}^N \) and \( \hat{s} \in \mathbb{R}^N \) of the mean and the covariance above evaluated on each observation can be computed in \( O(N) \) as

\[
\hat{m} = K_{X, \alpha} a \\
\hat{s} = \text{diag} (K_{X} - K_{X, \beta} (B^{-1} + K_\beta)^{-1} K_\beta, x) \\
= \text{diag}(K_X) - (K_{X, \beta} \circ (K_{X, \beta} (B^{-1} + K_\beta)^{-1})) 1 \\
= \text{diag}(K_X) - (K_{X, \beta} \circ (K_{X, \beta} B(I + K_\beta B)^{-1}) 1.
\]

Given \( \hat{m} \) and \( \hat{s} \), the expected log-likelihood can be evaluated either in closed-form for Gaussian likelihood or by sampling for general likelihoods.

#### A.2.2 Gradients

The computation of the gradients of the expected log-likelihood can done in two steps. First, we compute the gradients of \( \text{KL}[q||p] \) with respect to \( \hat{m} \), \( \hat{s} \), and \( \hat{s} \). Because \( \log p_0(y|f) \) is the sum of \( N \) terms, this step can be done in \( O(N) \): for each observation \( x \), let \( q(f(x)) = \mathcal{N}(f(x)|\hat{m}, \hat{s}) \) be a scalar Gaussian; under standard regularity conditions, we have

\[
\begin{align*}
\nabla_{\hat{m}} \text{KL} &= \text{E}_q[\log p_0 (y|f(x))] \\
\nabla_{\hat{s}} \text{KL} &= \text{E}_q[\nabla_{\hat{m}} \log q(f(x)) \log p_0 (y|f(x))] \\
\nabla_{\hat{e}} \text{KL} &= \text{E}_q[\nabla_{\hat{e}} \log q(f(x)) \log p_0 (y|f(x))]
\end{align*}
\]

where \( \nabla_{\hat{e}} \log p_0 (y|f(x)) \) can be found, for example, in (22). The above can be calculated in closed-form for Gaussian likelihood or by sampling for general likelihoods.

Next we propagate these gradients by chain rule. The results are summarized below.

---

\(^{[13]}\)The additional factor of 2 is due to \( K_\beta \) is symmetric.
Proposition 3. Let $e = E_q[\log p_d(y|f)]$. Suppose $k(x, x') = \rho^2 g_s(x, x')$ for some hyper-parameters $\rho, s \in \mathbb{R}$. The gradients of $e$ are as follows:

\[
\begin{align*}
\nabla_a e &= K_{X,\alpha}^T \nabla m e \\
\nabla_\rho e &= \text{diag}(a) \partial K_{X,\alpha}^T \nabla m e \\
\nabla_B e &= -(I + KB)^{-1} K_{X,\beta} \text{diag}(\nabla_\rho e) K_{X,\beta}(I + BK)^{-1} \\
\nabla_{\beta} \hat{e} &= 2(\partial K_{\beta}^T \odot (\Omega \text{diag}(\nabla_\rho e) \Omega^T)) (1 - 2(\Omega \odot \partial K_{\beta,\Sigma}) \nabla \rho e) \\
\nabla_{\log \rho} e &= \hat{m}^T \nabla m e + 2\hat{s}^T \nabla \rho e \\
\nabla_\rho e &= (\partial_\rho K_{X,\alpha} a)^T \nabla m e - 21^T (\Omega \odot \partial K_{\beta,\Sigma}) \nabla \rho e
\end{align*}
\]

where $\Omega = B(I + KB)^{-1} K_{\beta,\Sigma}$.

The derivation of Proposition 3 is only technical, so we omit it here.

B Experiment Setup

B.1 The Covariance Function

For all the models, we assume the prior is zero mean and has covariance defined by a SE-ARD kernel [19]

\[ k(x, x') = \rho^2 \psi_x^T \phi_x = \rho^2 \prod_{d=1}^D \exp \left( -\frac{(x_d - x'_d)^2}{2\sigma_d^2} \right), \]

where $\sigma_d > 0$ is the length scale of dimension $d$. For the variational posterior, we use the generalized SE-ARD kernel [2]

\[ \psi_x^T \psi_{x'} = \prod_{d=1}^D \left( \frac{2l_{x,d}l_{x',d}}{l_{x,d}^2 + l_{x',d}^2} \right)^{1/2} \exp \left( -\frac{\|x_d - x'_d\|^2}{l_{x,d}^2 + l_{x',d}^2} \right), \quad (15) \]

where $l_{x,d} = s_{x,d} \cdot c_{x,d}$ is the length-scale parameter. That is, we evaluate

\[ C[L_m f(\tilde{x}_m), L_n f(\tilde{x}_n)] = \psi_{x_m}^T \psi_{x_n} \]

where the associated length-scale parameters implicitly define the linear operators $L_m$ and $L_n$.

This kernel is first introduced in [26] by convoluting a SE-ARD kernel with Gaussian integral kernels, and later modified into its current form [15] in [2]. From [15], we see it contains SE-ARD as a special case. That is, $\psi_x = \phi_x$ when $c_{x,d} = 1, \forall d \in \{1, \ldots, D\}$. But in general $c_{x,d}$ can be a function of $x$. Therefore, it can be shown that $\psi_x$ spans an RKHS that contains the RKHSs spanned by $\phi_x$ for all length-scales, and every cross covariance can be computed as $C[L_m f(\tilde{x}_m), f(x)] = \rho \psi_{x_m}^T \phi_x$.

Note: all the algorithms in our comparisons use this generalized SE-ARD kernel.

B.2 Online Learning Procedure

Algorithm 1 summarizes the online learning procedure used by all stochastic algorithms (the algorithms differ only in whether the bases are shared and how the model is updated; see Table 1), where each learner has to optimize all the parameters on-the-fly using i.i.d. data. The hyper-parameters are first initialized heuristically by median trick using the first mini-batch (in the GPR experiments, $s_d$ is initialized as the median of pairwise distances of the sampled observations; $\sigma^2$ is initialized as the variance of the sampled outputs; $\rho = 1$). We incrementally build up the variational posterior by including $N_{\Delta} \leq N_m$ observations in each mini-batch as the initialization of new variational basis functions (we initialize a new variational basis as $\tilde{x}_m = x_n$ and $c_{x,d} = 1$, where $x_n$ is a sample from the current mini-batch). Then all the hyper-parameters and the variational parameters are updated online. These steps are repeated for $T$ iterations.


C  Complete Experimental Results

C.1  Experimental Results on KUKA datasets

<table>
<thead>
<tr>
<th></th>
<th>SVDGP</th>
<th>SVI</th>
<th>VSGPR</th>
<th>VSGPR</th>
<th>GPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y1</td>
<td>0.985</td>
<td>0.336</td>
<td>0.411</td>
<td>0.085</td>
<td>-3.840</td>
</tr>
<tr>
<td>Y2</td>
<td>1.359</td>
<td>0.458</td>
<td>0.799</td>
<td>0.468</td>
<td>-23.218</td>
</tr>
<tr>
<td>Y3</td>
<td>0.951</td>
<td>0.312</td>
<td>0.543</td>
<td>0.158</td>
<td>-8.145</td>
</tr>
<tr>
<td>Y4</td>
<td>1.453</td>
<td>0.528</td>
<td>0.906</td>
<td>0.722</td>
<td>-0.965</td>
</tr>
<tr>
<td>Y5</td>
<td>1.350</td>
<td>0.311</td>
<td>0.377</td>
<td>0.425</td>
<td>-0.990</td>
</tr>
<tr>
<td>Y6</td>
<td>1.278</td>
<td>0.367</td>
<td>0.631</td>
<td>0.559</td>
<td>-0.639</td>
</tr>
<tr>
<td>Y7</td>
<td>1.458</td>
<td>0.425</td>
<td>0.877</td>
<td>0.886</td>
<td>0.449</td>
</tr>
<tr>
<td>mean</td>
<td>1.262</td>
<td>0.391</td>
<td>0.649</td>
<td>0.472</td>
<td>-5.335</td>
</tr>
<tr>
<td>std</td>
<td>0.195</td>
<td>0.076</td>
<td>0.201</td>
<td>0.265</td>
<td>7.777</td>
</tr>
</tbody>
</table>

(a) Variational Lower Bound ($10^5$)

<table>
<thead>
<tr>
<th></th>
<th>SVDGP</th>
<th>SVI</th>
<th>VSGPR</th>
<th>VSGPR</th>
<th>GPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y1</td>
<td>0.058</td>
<td>0.186</td>
<td>0.165</td>
<td>0.171</td>
<td>0.257</td>
</tr>
<tr>
<td>Y2</td>
<td>0.028</td>
<td>0.146</td>
<td>0.095</td>
<td>0.126</td>
<td>0.249</td>
</tr>
<tr>
<td>Y3</td>
<td>0.058</td>
<td>0.195</td>
<td>0.133</td>
<td>0.181</td>
<td>0.298</td>
</tr>
<tr>
<td>Y4</td>
<td>0.027</td>
<td>0.124</td>
<td>0.088</td>
<td>0.114</td>
<td>0.198</td>
</tr>
<tr>
<td>Y5</td>
<td>0.028</td>
<td>0.195</td>
<td>0.178</td>
<td>0.132</td>
<td>0.243</td>
</tr>
<tr>
<td>Y6</td>
<td>0.034</td>
<td>0.178</td>
<td>0.137</td>
<td>0.140</td>
<td>0.224</td>
</tr>
<tr>
<td>Y7</td>
<td>0.028</td>
<td>0.155</td>
<td>0.099</td>
<td>0.108</td>
<td>0.146</td>
</tr>
<tr>
<td>mean</td>
<td>0.037</td>
<td>0.169</td>
<td>0.128</td>
<td>0.139</td>
<td>0.231</td>
</tr>
<tr>
<td>std</td>
<td>0.013</td>
<td>0.025</td>
<td>0.033</td>
<td>0.026</td>
<td>0.045</td>
</tr>
</tbody>
</table>

(b) Prediction Error (nMSE)

Table 3: Experimental results of KUKA1 after 2,000 iteration. $Y_i$ denotes the $i$th output.

<table>
<thead>
<tr>
<th></th>
<th>SVDGP</th>
<th>SVI</th>
<th>VSGPR</th>
<th>VSGPR</th>
<th>GPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y1</td>
<td>1.047</td>
<td>0.398</td>
<td>0.631</td>
<td>0.399</td>
<td>-3.709</td>
</tr>
<tr>
<td>Y2</td>
<td>1.387</td>
<td>0.450</td>
<td>0.767</td>
<td>0.515</td>
<td>-31.315</td>
</tr>
<tr>
<td>Y3</td>
<td>0.976</td>
<td>0.321</td>
<td>0.568</td>
<td>0.232</td>
<td>-12.230</td>
</tr>
<tr>
<td>Y4</td>
<td>1.404</td>
<td>0.307</td>
<td>0.630</td>
<td>0.634</td>
<td>-1.026</td>
</tr>
<tr>
<td>Y5</td>
<td>1.332</td>
<td>0.317</td>
<td>0.378</td>
<td>0.511</td>
<td>-0.340</td>
</tr>
<tr>
<td>Y6</td>
<td>1.260</td>
<td>0.368</td>
<td>0.585</td>
<td>0.538</td>
<td>-0.221</td>
</tr>
<tr>
<td>Y7</td>
<td>1.405</td>
<td>0.437</td>
<td>0.519</td>
<td>0.918</td>
<td>0.526</td>
</tr>
<tr>
<td>mean</td>
<td>1.259</td>
<td>0.400</td>
<td>0.583</td>
<td>0.538</td>
<td>-6.902</td>
</tr>
<tr>
<td>std</td>
<td>0.165</td>
<td>0.065</td>
<td>0.110</td>
<td>0.197</td>
<td>10.770</td>
</tr>
</tbody>
</table>

(a) Variational Lower Bound ($10^5$)

<table>
<thead>
<tr>
<th></th>
<th>SVDGP</th>
<th>SVI</th>
<th>VSGPR</th>
<th>VSGPR</th>
<th>GPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y1</td>
<td>0.056</td>
<td>0.168</td>
<td>0.126</td>
<td>0.151</td>
<td>0.281</td>
</tr>
<tr>
<td>Y2</td>
<td>0.026</td>
<td>0.147</td>
<td>0.102</td>
<td>0.124</td>
<td>0.248</td>
</tr>
<tr>
<td>Y3</td>
<td>0.056</td>
<td>0.194</td>
<td>0.127</td>
<td>0.179</td>
<td>0.325</td>
</tr>
<tr>
<td>Y4</td>
<td>0.029</td>
<td>0.127</td>
<td>0.127</td>
<td>0.110</td>
<td>0.186</td>
</tr>
<tr>
<td>Y5</td>
<td>0.029</td>
<td>0.189</td>
<td>0.170</td>
<td>0.125</td>
<td>0.232</td>
</tr>
<tr>
<td>Y6</td>
<td>0.035</td>
<td>0.181</td>
<td>0.144</td>
<td>0.144</td>
<td>0.232</td>
</tr>
<tr>
<td>Y7</td>
<td>0.034</td>
<td>0.152</td>
<td>0.166</td>
<td>0.104</td>
<td>0.133</td>
</tr>
<tr>
<td>mean</td>
<td>0.038</td>
<td>0.166</td>
<td>0.137</td>
<td>0.134</td>
<td>0.234</td>
</tr>
<tr>
<td>std</td>
<td>0.012</td>
<td>0.023</td>
<td>0.022</td>
<td>0.024</td>
<td>0.058</td>
</tr>
</tbody>
</table>

(b) Prediction Error (nMSE)

Table 4: Experimental results of KUKA2 after 2,000 iterations. $Y_i$ denotes the $i$th output.
### C.2 Experimental Results on MuJoCo datasets

#### Table 5: Experimental results of MUJOCO after 2,000 iterations. $Y_i$ denotes the $i$th output.

<table>
<thead>
<tr>
<th></th>
<th>SVGP</th>
<th>SVI</th>
<th>IVSGPR</th>
<th>VSGPR</th>
<th>GPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y_1$</td>
<td>7.373</td>
<td>3.195</td>
<td>5.948</td>
<td>4.312</td>
<td>-22.256</td>
</tr>
<tr>
<td>$Y_2$</td>
<td>6.019</td>
<td>2.141</td>
<td>3.905</td>
<td>2.328</td>
<td>-45.351</td>
</tr>
<tr>
<td>$Y_3$</td>
<td>6.350</td>
<td>2.543</td>
<td>4.695</td>
<td>2.991</td>
<td>-147.881</td>
</tr>
<tr>
<td>$Y_4$</td>
<td>5.852</td>
<td>2.417</td>
<td>4.792</td>
<td>2.468</td>
<td>-23.999</td>
</tr>
<tr>
<td>$Y_5$</td>
<td>6.280</td>
<td>2.699</td>
<td>5.316</td>
<td>3.622</td>
<td>-11296.669</td>
</tr>
<tr>
<td>$Y_6$</td>
<td>5.152</td>
<td>1.043</td>
<td>4.418</td>
<td>3.452</td>
<td>-8.626</td>
</tr>
<tr>
<td>$Y_7$</td>
<td>5.270</td>
<td>2.093</td>
<td>4.183</td>
<td>1.676</td>
<td>-7745.055</td>
</tr>
<tr>
<td>$Y_8$</td>
<td>6.471</td>
<td>2.585</td>
<td>5.040</td>
<td>3.068</td>
<td>-47.540</td>
</tr>
<tr>
<td>$Y_9$</td>
<td>5.293</td>
<td>0.979</td>
<td>2.592</td>
<td>1.482</td>
<td>-73477.168</td>
</tr>
</tbody>
</table>

| mean   | 6.007 | 2.178 | 4.543  | 2.822 | -10312.727  |
| std    | 0.673 | 0.692 | 0.898  | 0.871 | 22679.778   |

(a) Variational Lower Bound ($10^5$)

<table>
<thead>
<tr>
<th></th>
<th>SVGP</th>
<th>SVI</th>
<th>IVSGPR</th>
<th>VSGPR</th>
<th>GPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y_1$</td>
<td>0.049</td>
<td>0.087</td>
<td>0.067</td>
<td>0.088</td>
<td>0.133</td>
</tr>
<tr>
<td>$Y_2$</td>
<td>0.068</td>
<td>0.163</td>
<td>0.112</td>
<td>0.122</td>
<td>0.196</td>
</tr>
<tr>
<td>$Y_3$</td>
<td>0.064</td>
<td>0.134</td>
<td>0.091</td>
<td>0.112</td>
<td>0.213</td>
</tr>
<tr>
<td>$Y_4$</td>
<td>0.073</td>
<td>0.144</td>
<td>0.087</td>
<td>0.121</td>
<td>0.179</td>
</tr>
<tr>
<td>$Y_5$</td>
<td>0.068</td>
<td>0.132</td>
<td>0.080</td>
<td>0.103</td>
<td>0.159</td>
</tr>
<tr>
<td>$Y_6$</td>
<td>0.094</td>
<td>0.251</td>
<td>0.107</td>
<td>0.131</td>
<td>0.253</td>
</tr>
<tr>
<td>$Y_7$</td>
<td>0.084</td>
<td>0.168</td>
<td>0.100</td>
<td>0.145</td>
<td>0.348</td>
</tr>
<tr>
<td>$Y_8$</td>
<td>0.063</td>
<td>0.132</td>
<td>0.087</td>
<td>0.104</td>
<td>0.178</td>
</tr>
<tr>
<td>$Y_9$</td>
<td>0.088</td>
<td>0.255</td>
<td>0.165</td>
<td>0.131</td>
<td>0.258</td>
</tr>
</tbody>
</table>

| mean   | 0.072 | 0.163 | 0.099  | 0.118 | 0.213 |
| std    | 0.013 | 0.053 | 0.026  | 0.016 | 0.061 |

(b) Prediction Error (nMSE)

#### Table 6: Experimental results of MUJOCO after 2,000 iterations. $Y_i$ denotes the $i$th output.

<table>
<thead>
<tr>
<th></th>
<th>SVGP</th>
<th>SVI</th>
<th>IVSGPR</th>
<th>VSGPR</th>
<th>GPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y_1$</td>
<td>7.249</td>
<td>3.013</td>
<td>6.429</td>
<td>4.161</td>
<td>-33.219</td>
</tr>
<tr>
<td>$Y_2$</td>
<td>5.994</td>
<td>2.475</td>
<td>4.800</td>
<td>2.770</td>
<td>-23.276</td>
</tr>
<tr>
<td>$Y_3$</td>
<td>6.259</td>
<td>2.258</td>
<td>4.819</td>
<td>3.044</td>
<td>-59.757</td>
</tr>
<tr>
<td>$Y_4$</td>
<td>5.935</td>
<td>2.093</td>
<td>4.489</td>
<td>2.547</td>
<td>-27.259</td>
</tr>
<tr>
<td>$Y_5$</td>
<td>6.307</td>
<td>2.452</td>
<td>5.457</td>
<td>3.725</td>
<td>-1.786</td>
</tr>
<tr>
<td>$Y_6$</td>
<td>7.320</td>
<td>1.087</td>
<td>4.639</td>
<td>4.043</td>
<td>-24.198</td>
</tr>
<tr>
<td>$Y_7$</td>
<td>5.346</td>
<td>1.754</td>
<td>3.947</td>
<td>1.667</td>
<td>-255179.052</td>
</tr>
<tr>
<td>$Y_8$</td>
<td>6.448</td>
<td>2.505</td>
<td>5.193</td>
<td>3.812</td>
<td>-190.294</td>
</tr>
<tr>
<td>$Y_9$</td>
<td>6.237</td>
<td>0.683</td>
<td>2.596</td>
<td>2.241</td>
<td>-37673.328</td>
</tr>
</tbody>
</table>

| mean   | 6.350 | 2.036 | 4.708  | 3.112 | -32579.130  |
| std    | 0.586 | 0.699 | 0.993  | 0.825 | 79570.425   |

(a) Variational Lower Bound ($10^5$)

<table>
<thead>
<tr>
<th></th>
<th>SVGP</th>
<th>SVI</th>
<th>IVSGPR</th>
<th>VSGPR</th>
<th>GPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y_1$</td>
<td>0.051</td>
<td>0.095</td>
<td>0.056</td>
<td>0.085</td>
<td>0.138</td>
</tr>
<tr>
<td>$Y_2$</td>
<td>0.069</td>
<td>0.133</td>
<td>0.085</td>
<td>0.111</td>
<td>0.186</td>
</tr>
<tr>
<td>$Y_3$</td>
<td>0.066</td>
<td>0.149</td>
<td>0.087</td>
<td>0.113</td>
<td>0.182</td>
</tr>
<tr>
<td>$Y_4$</td>
<td>0.071</td>
<td>0.160</td>
<td>0.094</td>
<td>0.127</td>
<td>0.197</td>
</tr>
<tr>
<td>$Y_5$</td>
<td>0.065</td>
<td>0.137</td>
<td>0.074</td>
<td>0.101</td>
<td>0.148</td>
</tr>
<tr>
<td>$Y_6$</td>
<td>0.051</td>
<td>0.241</td>
<td>0.097</td>
<td>0.073</td>
<td>0.139</td>
</tr>
<tr>
<td>$Y_7$</td>
<td>0.081</td>
<td>0.187</td>
<td>0.107</td>
<td>0.142</td>
<td>0.363</td>
</tr>
<tr>
<td>$Y_8$</td>
<td>0.063</td>
<td>0.133</td>
<td>0.081</td>
<td>0.106</td>
<td>0.214</td>
</tr>
<tr>
<td>$Y_9$</td>
<td>0.067</td>
<td>0.270</td>
<td>0.157</td>
<td>0.106</td>
<td>0.300</td>
</tr>
</tbody>
</table>

| mean   | 0.065 | 0.167 | 0.093  | 0.107 | 0.207 |
| std    | 0.009 | 0.053 | 0.026  | 0.019 | 0.072 |

(b) Prediction Error (nMSE)