Subset-of-Data Variational Inference for Deep Gaussian-Processes Regression

Ayush Jain[1] **P. K. Srijith**[1] **Mohammad Emtiyaz Khan**[2]

[1]Department of Computer Science and Engineering , Indian Institute of Technology Hyderabad, India
[2]RIKEN Center for AI Project , Tokyo, Japan

Abstract

Deep Gaussian Processes (DGPs) are multi-layer, flexible extensions of Gaussian Processes but their training remains challenging. Sparse approximations simplify the training but often require optimization over a large number of inducing inputs and their locations across layers. In this paper, we simplify the training by setting the locations to a *fixed* subset of data and sampling the inducing inputs from a variational distribution. This reduces the trainable parameters and computation cost without significant performance degradations, as demonstrated by our empirical results on regression problems. Our modifications simplify and stabilize DGP training while making it amenable to sampling schemes for setting the inducing inputs.

1 INTRODUCTION

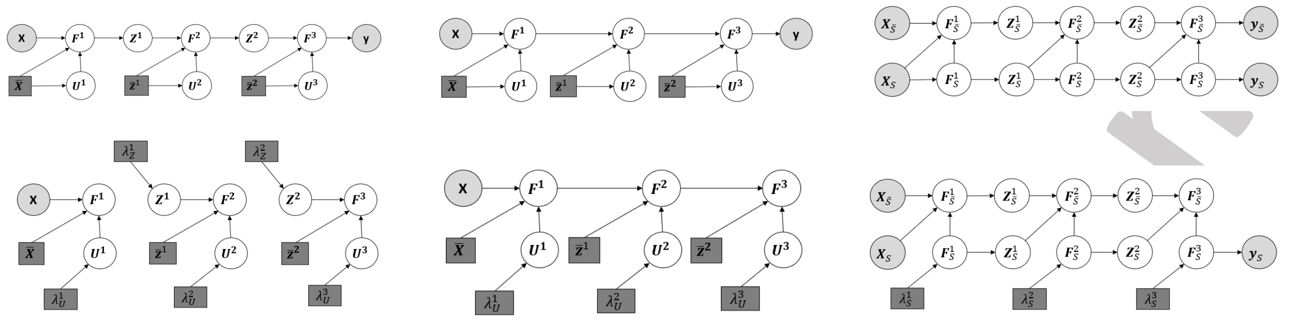
Deep Gaussian Processes (DGPs) aim to extend the functional-learning capabilities of Gaussian Processes (GPs) to improve their flexibility. A DGP consists of multiple layers of GPs stacked one over the other [Damianou and Lawrence, 2013, Damianou, 2015] which enables us to model complex functions, such as non-stationary and discontinuous functions. DGPs are motivated by the deep architectures used in deep learning but they promise to overcome the limitations of deep learning, e.g., they can deal with smaller datasets, improve uncertainty estimates, and perform model selection.

Unfortunately, obtaining good performance with DGP on large problems remains a challenge, mainly due the inefficient training procedures currently used. Unlike deep learning, the training procedures for DGPs are computationally expensive, slow to run, and rarely beat the performance of deep learning. Posterior inference is even more challenging than GPs and involve multiple large matrix inversions of size $O(N^3)$ where N is the number of data examples. Existing methods mitigate such difficulties by using *sparse variational inference* (VI) [Titsias, 2009] which augment the model with auxiliary variables called *inducing inputs*. An example is shown in Fig. 1(a) (top row) where inducing input $\bar{\mathbf{Z}}^l$ and corresponding functions $\mathbf{U}^l$ are introduced in every layer l. These auxiliary quantities are different from the functions and inputs defined over input $\mathbf{X}$ in the training data (denoted by $\mathbf{F}^l$ and $\mathbf{Z}^l$ respectively). During inference, all the auxiliary quantities need to be estimated along with the variational distribution parameters and GP kernel hyperparameters to obtain a posterior approximation. Due to this, the number of trainable parameters is often very large.

Various strategies for training have been proposed in the past. The approach of Damianou and Lawrence [2013] (Fig. 1(a)) assumes a mean-field variational approximation over $\mathbf{Z}^l$ and $\mathbf{U}^l$, and estimates their variational parameters (denoted by $\boldsymbol{\lambda}_Z^l$ and $\boldsymbol{\lambda}_U^l$ respectively), as well as the inducing inputs $\bar{\mathbf{Z}}^l$. All these trainable parameters are shown with square gray boxes in the figure. The doubly stochastic variational inference (DSVI) approach of Salimbeni and Deisenroth [2017], shown in Fig. 1(b), reduces the number of trainable parameters by integrating $\mathbf{Z}^l$ out. The approach still needs to estimate the variational parameters $\boldsymbol{\lambda}_U^l$ and inducing inputs $\bar{\mathbf{Z}}^l$. Our goal in this paper is to further reduce the number of trainable parameters.

Our key idea to reduce the number of parameters is to use a fixed subset of data as inducing inputs, i.e., a subset $S \subset \{1, 2, \dots, N\}$ of size M with $M \ll N$. Instead of using auxiliary inputs $\bar{\mathbf{Z}}^l$ and functions $\mathbf{U}^l$, we use the inputs $\mathbf{Z}_S^l$ and functions $\mathbf{F}_S^l$ defined over the subset S; see the top row in Fig. 1(c). We introduce variational distributions over $\mathbf{F}_S^l$ with parameters denoted by $\boldsymbol{\lambda}_S^l$ while keeping the relationship between the rest of the quantities as defined by the DGP model (see the bottom row in Fig. 1(c)). The trainable parameters are $\boldsymbol{\lambda}_S^l, \forall l$, the size of which is strictly lower than the methods in Fig. 1(a) and 1(b) (assuming the number of inducing inputs M to be the same).

Accepted for the 37th Conference on Uncertainty in Artificial Intelligence (UAI 2021).



(a) Damianou and Lawrence [2013] (b) Salimbeni and Deisenroth [2017] (c) Our Subset-of-Data Method

Figure 1: Top row shows the (augmented) DGP model for input $\mathbf{X} \in \mathbb{R}^{N \times H}$ and output $\mathbf{y} \in \mathbb{R}^{N}$ for three methods, and bottom row shows the corresponding approximate posterior. Trainable parameters are shown with square-gray boxes in the bottom row, and we can see that our subset-of-data method requires the lowest number of parameters. Methods shown in the top row of (a) and (b) both use an augmented DGP model with inducing inputs $\bar{\mathbf{Z}}^l \in \mathbb{R}^{M \times D^l}$ and function values $\mathbf{U}^l \in \mathbb{R}^{M \times D^l}$. As a consequence, they require estimation of variational parameters $\boldsymbol{\lambda}_U^l$ for the variational distribution $q(\mathbf{U}^l)$ and inducing inputs $\bar{\mathbf{Z}}^l$, as shown in the bottom row. The method in (a) additionally requires variational parameters for $\mathbf{Z}^l$. Our method shown in (c) replaces $\mathbf{U}^l$ by the functions $\mathbf{F}_S^l \in \mathbb{R}^{M \times D^l}$ defined over a subset S of M data examples (see the top row; $\bar{S}$ denotes the set of training examples other than those in S). As shown in the bottom row, we only need to learn the parameters $\boldsymbol{\lambda}_S^l$ of $q(\mathbf{F}_S^l)$, since the rest of the quantities follow the same relationship as the DGP model. This leads to a reduction in the number of trainable parameters. Additionally, the inputs $\mathbf{Z}^l$ are not learned but sampled given $\mathbf{F}^l$.

Our approach exploits the labels $\mathbf{y}_S$ associated with the inducing input $\mathbf{X}_S$ to form the variational posterior and consequently simplifying the computation of the evidence lower-bound. The resulting bound naturally combines the predictive likelihood and marginal likelihood together leading to a simple yet effective solution (see (7) in Section 3). The idea of choosing inducing inputs from training set simplifies the inference due to the associated labels, ultimately leading to a reduction in the number of variational parameters. The empirical results suggest that our method reduces the cost of DGP inference without a significant degradation in the performance. An additional advantage of our method is that the inducing inputs $\mathbf{Z}_S^l$ are not trained but sampled using the variational distribution over $\mathbf{F}_S^l$.

In our experiments, we fix the subset S by clustering the data before training, but our methods is amenable to sampling approaches, such as leverage score [Alaoui and Mahoney, 2015] and determinantal point process [Kathuria et al., 2016], where the subsets can be sampled during training. We expect our approach to further improve when augmented with such sampling approaches. Throughout the paper, we focus primarily on improving the methods of Damianou and Lawrence [2013] and Salimbeni and Deisenroth [2017], but our approach could potentially be useful for many other variants, e.g., methods using amortized inference [Dai et al., 2016], nested inference [Hensman and Lawrence, 2014], approximate Expectation Propagation [Bui et al., 2016], random Fourier features expansion [Cutajar et al., 2017] and implicit posterior variational inference [Yu et al., 2019]. We only address regression problems, but extension to classification and multi-output DGPs can be obtained by using standard non-conjugate inference procedures.

2 DEEP GAUSSIAN PROCESSES

In this section we provide a background on deep Gaussian process models and their training methods. We consider the regression problem with N training data points, $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ and the corresponding labels $\mathbf{y} = \{y_n\}_{n=1}^N$, where $\mathbf{x}_n \in \mathcal{R}^H$ and $y_n \in \mathcal{R}$. We assume that there exists a regression function $f : \mathcal{R}^H \to \mathcal{R}$ which maps the training data to outputs, and our goal is to learn the function.

Deep Gaussian Processes (DGPs) provide a rich and flexible prior to model functions by stacking several GP priors. The DGP model described in Damianou and Lawrence [2013], Damianou [2015] (shown in Fig. 2) models the regression function by as a composition of several layers of functions, $\mathbf{y} = \mathbf{f}^L \circ (\mathbf{f}^{L-1} \ldots \circ (\mathbf{f}^1(\mathbf{X})))$(assuming L layers). The l^{th} layer consists of D^l functions $f^l = \{f_d^l\}_{d=1}^{D^l}$ mapping representations in layer $l-1$ to obtain D^l dimensional representation for layer l. In each layer independent GP priors are placed over the functions f_d^l conditioned on the output of the previous layer,

$$f_d^l(\cdot) \sim \mathcal{GP}(\mu_d^l(\cdot), k^l(\cdot, \cdot)), \tag{1}$$

where $\mu_d^l : \mathcal{R}^{D^{l-1}} \to \mathcal{R}$ is the mean function and $k^l : \mathcal{R}^{D^{l-1}} \times \mathcal{R}^{D^{l-1}} \to \mathcal{R}$ is the covariance function. The functions $f_d^1(\cdot)$ in the first layer act on the inputs $\mathbf{x}_n$ to produce

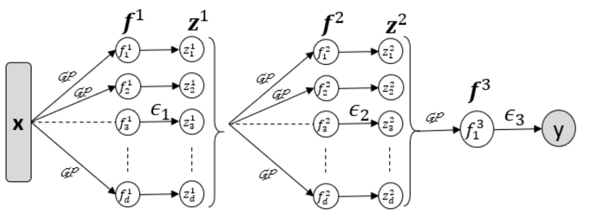


Figure 2: Deep Gaussian Process Model models a scalar output y given an input vector $\mathbf{x}$ with nested layers of function $\mathbf{f}^l$, generated using GPs, and their noise versions $\mathbf{z}^l$.

the mapping $F^1_{n,d} := f^1_d(\mathbf{x}_n)$. The first-layer representations Z^1_n are obtained by adding noise to these mappings, $Z^1_{n,d} = F^1_{n,d} + \epsilon_1$, where $\epsilon_1 \sim \mathcal{N}(0, \sigma_1^2)$. These are then fed as inputs to the next layer and the process is repeated. We use $\mathbf{Z}^l, \mathbf{F}^l \in \mathbb{R}^{N \times D^l}$ to denote the matrices obtained with entries $Z^l_{n,d}$ and $F^l_{n,d}$ respectively.

The joint distribution of $\mathbf{y}$ and $\mathbf{F}^l, \mathbf{Z}^l$ over all layers is

$$p(\mathbf{y}, \mathbf{F}^L, \mathbf{Z}^{L-1}, \ldots, \mathbf{F}^2, \mathbf{Z}^1, \mathbf{F}^1|\mathbf{X}) \tag{2}$$
$$= \underbrace{\prod_{n=1}^{N} p(y_n|F_n^L)}_{\text{Likelihood}} \underbrace{\left[\prod_{l=1}^{L-1} p(\mathbf{F}^{l+1}|\mathbf{Z}^l) p(\mathbf{Z}^l|\mathbf{F}^l)\right] p(\mathbf{F}^1|\mathbf{X})}_{\text{Deep GP Prior}}$$

Here, the likelihood over the observation $p(y_n|F_n^L) = \mathcal{N}(y_n; F_n^L, \sigma_L^2)$, and the factors associated with intermediate layers can be factorized as

$$p(\mathbf{Z}^l|\mathbf{F}^l) = \prod_{n=1}^{N} \prod_{d=1}^{D^l} \mathcal{N}(Z^l_{n,d}; F^l_{n,d}, \sigma_l^2) \tag{3}$$

The prior over the functions at some layer l and dimension d is considered to be a zero mean GP with kernel $k^l(\cdot,\cdot)$, i.e. $f^l(\cdot) = GP(0, k^l(\cdot,\cdot))$. Consequently, the function values over the data points are distributed as a zero mean Gaussian.

$$p(F^l_{:,d}|\mathbf{Z}^{l-1}) = \mathcal{N}(0, \mathbf{K}^l_{Z^{l-1},Z^{l-1}})$$

Here, $\mathbf{K}^l_{Z^{l-1},Z^{l-1}}$ is an $N \times N$ co-variance matrix obtained by evaluating the kernel $k^l(\cdot,\cdot)$ on the $l-1$ layer latent representations of all the data points. The priors and likelihood are conditioned on kernel hyper-parameters and noise variance(σ_l). Squared exponential kernel is most widely used kernel in this setting which have length scales and variance as hyperparameters. Hyper-parameter learning requires optimisation of marginal likelihood which is intractable in Deep GPs. This is because latent representation appear nonlinearly in the full likelihood due to the form of $p(\mathbf{F}^{l+1}|\mathbf{Z}^l)$ terms. Due to underlying intractability, approximate Bayesian methods are used to compute an approximate inference for Deep GPs.

Training Deep GPs require approximate inference methods which largely relies on sparse variational-inference methods [Damianou and Lawrence, 2013, Salimbeni and Deisenroth, 2017]. These methods compute a tractable lower bound for marginal likelihood using sparse GPs [Titsias, 2009, Hensman et al., 2013]. These approaches simultaneously addresses intractability and scalability issues in Deep GPs by introducing inducing points with inducing input $\bar{\mathbf{Z}}^l$ and output $\mathbf{U}^l$ for each layer l, all of which are learnt from the variational lower bound (top figure in Fig. 1(a)). The posterior inference is simplified by estimating $\bar{\mathbf{Z}}^l$ and introducing a Gaussian approximation $q(\mathbf{U}^l)$ with variational parameters $\boldsymbol{\lambda}^l_U$. Given these two quantities the cost of posterior inference reduces drastically. The corresponding approximate posterior introduced in Damianou and Lawrence [2013] is shown at the bottom figure in Fig. 1(a) where an additional variational distribution over $\mathbf{Z}^l$ is used.

Even though the inference is simplified with the method of Damianou and Lawrence [2013], the number of variational parameters is still quite large. Salimbeni and Deisenroth [2017] reduce the number by marginalizing $\mathbf{Z}^l$ thereby not requiring the additional distribution over it. During inference, $\mathbf{Z}^l$ is obtained by using a forward sampling from the DGP model. The approach Salimbeni and Deisenroth [2017] reduces the cost of inference by reducing the number of trainable parameters.

3 SUBSET-OF-DATA VARIATIONAL INFERENCE (SOD-VI)

Our goal in this paper is to reduce the number of trainable parameters. We view the inducing inputs $\mathbf{U}^l$ as inputs without any real inputs $\mathbf{x}$ or labels y. Our key idea then is to replace them by latent functions $\mathbf{F}^l_S$ defined over a subset S of the training data. This reduces the number of trainable parameters, because the input locations $\bar{\mathbf{Z}}^l$ are replaced by $\mathbf{Z}^l_S$ which can be sampled from the DGP model, instead of being learned. We first describe this for a single-layer GP.

3.1 SUBSET-OF-DATA VI FOR GP

We first consider deriving the subset-of-data VI for a simple one layer GP model with $\mathbf{F}$ representing function values at data points $\mathbf{X}$. We write the joint likelihood over observations (outputs) and latent function values as

$$p(\mathbf{y}, \mathbf{F}) = \left[\prod_{n=1}^{N} p(y_n|F_n)\right] p(\mathbf{F})$$
$$= p(\mathbf{y}_S|\mathbf{F}_S) p(\mathbf{y}_{\bar{S}}|\mathbf{F}_{\bar{S}}) p(\mathbf{F}_{\bar{S}}|\mathbf{F}_S) p(\mathbf{F}_S) \tag{4}$$

where $S \subset \{1, 2, \ldots, N\}$ is a subset of data examples, $\mathbf{F}_S$ denotes the latent function values over the inputs indexed by S and $\mathbf{y}_S$ are the corresponding observations. $\bar{S}$ is the

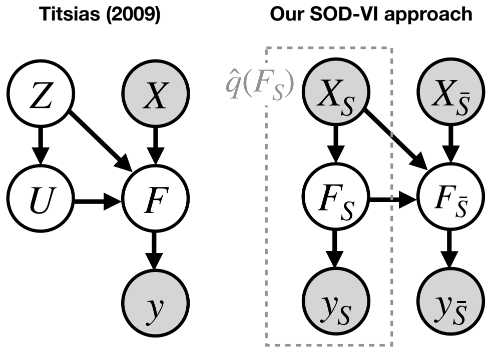


Figure 3: Titsias [2009]'s approach uses unknown input locations $\mathbf{Z}$ to define inducing inputs $\mathbf{U}$. In contrast, we use *known* locations $\mathbf{X}_S$ from the training data. Since the labels $\mathbf{y}_S$ for these locations are available, the posterior over $\mathbf{F}_S$ is also known and fixed (see (6) for an expression). Our lower bound however has the same form as the one traditionally used in sparse variational GP methods (see (7)).

complement of subset S (indexes of data examples not in S), and $\mathbf{F}_{\bar{S}}$ and $\mathbf{y}_{\bar{S}}$ denote the corresponding latent function values and observations, respectively. The subset S can be chosen randomly, by clustering the training data and choosing data points closest to the centroids, or by sampling approaches such as leverage score and determinantal point processes. Throughout, we use the clustering approach.

Our goal is to obtain a posterior over $\mathbf{F} = \{\mathbf{F}_{\bar{S}}, \mathbf{F}_S\}$, while using $\mathbf{F}_S$ as the inducing points and the corresponding $\mathbf{X}_S$ as the inducing inputs. To build a variational approximation, we first rewrite the posterior over $\mathbf{F}$ as follows,

$$p(\mathbf{F}|\mathbf{y}) \propto p(\mathbf{y}_{\bar{S}}|\mathbf{F}_{\bar{S}})p(\mathbf{F}_{\bar{S}}|\mathbf{F}_S)\underbrace{\frac{p(\mathbf{y}_S|\mathbf{F}_S)p(\mathbf{F}_S)}{p(\mathbf{y}_S)}}_{=p(\mathbf{F}_S|\mathbf{y}_S)} \quad (5)$$

where we have explicitly grouped the terms corresponding to the posterior $p(\mathbf{F}_S|\mathbf{y}_S)$. We define a variational distribution which has the same structure but use a variational approximation $q(\mathbf{F}_S)$ instead of the prior $p(\mathbf{F}_S)$,

$$p(\mathbf{F}|\mathbf{y}) \approx \frac{1}{\hat{Z}}p(\mathbf{F}_{\bar{S}}|\mathbf{F}_S)\underbrace{\frac{p(\mathbf{y}_S|\mathbf{F}_S)q(\mathbf{F}_S)}{p(\mathbf{y}_S)}}_{=\hat{q}(\mathbf{F}_S)}, \quad (6)$$

where $\hat{Z}$ is the normalizing constant. The distribution $\hat{q}(\mathbf{F}_S)$ is an approximation to the posterior $p(\mathbf{F}_S|\mathbf{y}_S)$, obtained via $q(\mathbf{F}_S)$. The corresponding posterior over $\mathbf{F}_{\bar{S}}$ can be obtained by marginalizing over $\mathbf{F}_S$, which we denote by $q(\mathbf{F}_{\bar{S}}) = \int p(\mathbf{F}_{\bar{S}}|\mathbf{F}_S)\hat{q}(\mathbf{F}_S)d\mathbf{F}_S$.

The above approach is different from the standard variational approach of Titsias [2009] where the inducing inputs do not have any labels associated with them, and we are forced to use $p(\mathbf{F}|\mathbf{y}) \approx p(\mathbf{F}_{\bar{S}}|\mathbf{F}_S)\hat{q}(\mathbf{F}_S)/\hat{Z}$ with an arbitrary free-form Gaussian $\hat{q}(\mathbf{F}_S)$. In our case, due to the associated label, $\hat{q}(\mathbf{F}_S) = p(\mathbf{F}_S|\mathbf{y}_S)$ which is obtained by choosing $q(\mathbf{F}_S) = p(\mathbf{F}_S)$. Our approach results in a simpler variational posterior than the approach of Titsias [2009]. See Fig. 3 for an illustration.

We will now show that, for the defined $\hat{q}(\mathbf{F}_S)$ and $q(\mathbf{F}_{\bar{S}})$, the variational lower bound can be derived as

$$\begin{aligned} &E_{p(\mathbf{F}_{\bar{S}}|\mathbf{F}_S)\hat{q}(\mathbf{F}_S)}\left[\log\left(\frac{p(\mathbf{y}|\mathbf{F})p(\mathbf{F}_{\bar{S}}|\mathbf{F}_S)p(\mathbf{F}_S)}{p(\mathbf{F}_{\bar{S}}|\mathbf{F}_S)\hat{q}(\mathbf{F}_S)}\right)\right] \\ &= E_{q(\mathbf{F}_{\bar{S}})}[\log(p(\mathbf{y}_{\bar{S}}|\mathbf{F}_{\bar{S}})] + E_{\hat{q}(\mathbf{F}_S)}[\log(p(\mathbf{y}_S|\mathbf{F}_S)] \\ &\qquad - KL(\hat{q}(\mathbf{F}_S)\|p(\mathbf{F}_S)). \quad (7)\end{aligned}$$

This is similar to the lower bounds used in Titsias [2009] and Hensman et al. [2013] but with $\mathbf{F}_S$ as the inducing points and the distribution set to $\hat{q}(\mathbf{F}_S)$. In our case, the distribution exploits the structure shown in (6) by using the labels, but for Titsias [2009] this can be an arbitrary Gaussian. The advantage with our approach is that we do not need to estimate the inducing inputs since they are fixed at the inputs $\mathbf{X}_S$. Note that in general our approach will give different results than [Titsias, 2009] due to the choice of the inducing inputs as well as the variational approximation. We will show in the next section that our approach is useful in reducing number of trainable parameters for Deep GPs.

The variational lower bound (7) takes an interesting form where the first term is the negative log-predictive probability (NLPP) [Shevade and Sundararajan, 2009] on a validation set $\bar{S}$, and the rest of the terms only depend on S. Therefore, we can see the lower bound maximization as maximizing the fit over the data both in S and $\bar{S}$, which is a better approach than the classical NLPP based approaches considering $\bar{S}$ alone. The variational parameters $\boldsymbol{\lambda}_S$ associated with $q(\mathbf{F}_S)$ (for instance, mean and Covariance parameters associated with a variational Gaussian approximation) and hyper-parameters such as kernel parameters are learnt by maximizing the variational lower bound.

3.2 SUBSET-OF-DATA VI FOR DGP

We will now extend the subset-of-data approach to DGPs. Fig. 1(c) (top row) shows the graphical model of the DGP model, where we have split the variables into two sets corresponding to examples in S and $\bar{S}$. Comparing to the other methods in Fig. 1, in our approach the subset S can be seen as playing the role of the inducing points. Consequently, the joint distribution also can be written in a conditional form where the term over S are conditioned on the terms over $\bar{S}$ as shown below:

$$\text{Likelihood: } p(\mathbf{y}_S|\mathbf{F}_S^L)p(\mathbf{y}_{\bar{S}}|\mathbf{F}_{\bar{S}}^L) \quad (8)$$

$$\begin{aligned}\text{DGP Prior: } &\left[\prod_{l=L-1}^{1} p(\mathbf{F}_{\bar{S}}^{l+1}|\mathbf{F}_S^{l+1},\mathbf{Z}^l)p(\mathbf{F}_S^{l+1}|\mathbf{Z}_S^l)p(\mathbf{Z}^l|\mathbf{F}^l)\right] \\ &\times p(\mathbf{F}_{\bar{S}}^1|\mathbf{F}_S^1,\mathbf{X})p(\mathbf{F}_S^1|\mathbf{X}_S) \quad (9)\end{aligned}$$

Similarly to the single-layer GP case, we can simply replace the priors $p(\mathbf{F}_S^{l+1}|\mathbf{Z}_S^l)$ by $q(\mathbf{F}_S^{l+1})$ and keep the rest of the structure same as the original model, to get the following variational approximation:

$$\begin{aligned}
&\bar{q}(\mathbf{F}^{1:L}, \mathbf{Z}^{1:L-1}) \propto \\
&p(\mathbf{y}_S|\mathbf{F}_S^L)\left(\prod_{l=L-1}^{1} p(\mathbf{F}_{\bar{S}}^{l+1}|\mathbf{F}_S^{l+1}, \mathbf{Z}^l)q(\mathbf{F}_S^{l+1})p(\mathbf{Z}^l|\mathbf{F}^l)\right) \\
&\qquad \times p(\mathbf{F}_{\bar{S}}^1|\mathbf{F}_S^1, X)q(\mathbf{F}_S^1). \qquad (10)
\end{aligned}$$

where where $q(\mathbf{F}_S^l)$ represents the variational distribution over $\mathbf{F}_S^l$, which is typically a Gaussian with mean and Covariance as variational parameters, $\boldsymbol{\lambda}_S^l = \{\boldsymbol{\mu}_S^l, \boldsymbol{\Sigma}_S^l\}$. The marginals and conditions derived from the above distributions will be denoted by $q(\cdot)$ to simplify the notation, e.g., $q(\mathbf{Z}_S^l)$ will be the marginals of $\mathbf{Z}_S^l$ obtained according to the graphical model shown in Fig. 1(c) (bottom row).

There are two main points to note. First, in the graph structure of the posterior approximation shown in Fig. 1(c) (bottom row), we see that the directed arrow from $\mathbf{Z}^l$ to $\mathbf{F}^l$ is removed. Second, the inducing inputs $\mathbf{Z}_S^l$ can be obtained directly from the samples $\mathbf{F}_S^l$ of the variational distribution. Unlike previous approaches, we do not need to learn them. This is the main reason behind the reduction in the number of trainable parameters with our approach.

Given the posterior approximation, the variational lower bound is directly obtained by using (8), (9) and (10),

$$\begin{aligned}
L &= E_{\bar{q}\left(\mathbf{F}^{1:L}, \mathbf{Z}^{1:L-1}\right)} \log\left(\frac{p(\mathbf{y}, \mathbf{F}^{1:L}, \mathbf{Z}^{1:L-1}|\mathbf{X})}{\bar{q}\left(\mathbf{F}^{1:L}, \mathbf{Z}^{1:L-1}\right)}\right) \\
&= E_{q(\mathbf{F}_{\bar{S}}^L)}[\log(p(\mathbf{y}_{\bar{S}}|\mathbf{F}_{\bar{S}}^L))] + E_{\hat{q}(\mathbf{F}_S^L)}[\log(p(\mathbf{y}_S|\mathbf{F}_S^L))] \\
&\quad - \sum_{d=1}^{D^1} KL(q(\mathbf{F}_{S,d}^1)\|p(\mathbf{F}_{S,d}^1|\mathbf{X}_S)) \\
&\quad - \sum_{l=2}^{L-1}\sum_{d=1}^{D^l} E_{q(\mathbf{Z}_S^{l-1})}\left[KL(q(\mathbf{F}_{S,d}^l)\|p(\mathbf{F}_{S,d}^l|\mathbf{Z}_S^{l-1}))\right] \\
&\quad - E_{q(\mathbf{Z}_S^{L-1})}[KL(\hat{q}(\mathbf{F}_S^L)\|p(\mathbf{F}_S^L|\mathbf{Z}_S^{L-1}))]. \qquad (11)
\end{aligned}$$

The form of the lower bound is slightly more complicated than (7), but the last three lines here are simply an expansion of the KL term splitting over the layers and dimensions. Another minor difference is that expectation with respect to $q(\mathbf{Z}_S^l)$ needs to be taken in the third line. We also note that the derivation can be easily extended to the multi-output case, where the first two terms and the last term in (11) will have an additional summation over the multiple outputs.

We now derive quantities required to compute the lower bound. We assume a Gaussian variational approximation $q(\mathbf{F}_S^L) = \mathcal{N}(\mathbf{F}_S^L; \boldsymbol{\mu}_S^L, \boldsymbol{\Sigma}_S^L)$ for the last layer and $q(\mathbf{F}_{S,d}^{l-1}) := \mathcal{N}(\mathbf{F}_{S,d}^{l-1}; \boldsymbol{\mu}_{S,d}^{l-1}, \boldsymbol{\Sigma}_{S,d}^{l-1})$ for the layer $l-1$, using which we derive $\hat{q}(\mathbf{F}_S^L)$, required in the second and last term in (11),

$$\begin{aligned}
\hat{q}(\mathbf{F}_S^L) &= \frac{p(\mathbf{y}_S|\mathbf{F}_S^L)q(\mathbf{F}_S^L)}{\mathcal{Z}} = \mathcal{N}(\hat{\boldsymbol{\mu}_S^L}, \hat{\boldsymbol{\Sigma}_S^L}) \qquad (12) \\
\text{where } \hat{\boldsymbol{\Sigma}_S^L} &= ((\boldsymbol{\Sigma}_S^L)^{-1} + (\sigma^2\boldsymbol{I})^{-1})^{-1} \\
\hat{\boldsymbol{\mu}_S^L} &= \hat{\boldsymbol{\Sigma}_S^L}((\sigma^2\mathbf{I})^{-1}\mathbf{y}_S + (\boldsymbol{\Sigma}_S^L)^{-1}\boldsymbol{\mu}_S^L)
\end{aligned}$$

as well as $q(\mathbf{Z}_S^{l-1})$ required in the last two terms in (11),

$$\begin{aligned}
q(\mathbf{Z}_S^{l-1}) &= \prod_{d=1}^{D^{l-1}} \int p(\mathbf{Z}_{S,d}^{l-1}|\mathbf{F}_{S,d}^{l-1})q(\mathbf{F}_{S,d}^{l-1})d\mathbf{F}_{S,d}^{l-1} \qquad (13) \\
&= \prod_{d=1}^{D^{l-1}} \mathcal{N}(\mathbf{Z}_{S,d}^{l-1}|\boldsymbol{\mu}_{S,d}^{l-1}, \sigma^2\boldsymbol{I} + \boldsymbol{\Sigma}_{S,d}^{l-1}). \qquad (14)
\end{aligned}$$

We also need the marginal distribution $q(\mathbf{F}_{\bar{S}}^L)$ to compute the expectation in the first term in (11). This can be obtained by integrating out all the latent variables except $\mathbf{F}_{\bar{S}}^L$ in the variational posterior approximation (10). The latent function values in the intermediate layers $\mathbf{F}^l$ can be integrated out and consequently the marginal can be written as [1]

$$\int q(\mathbf{F}_{\bar{S}}^L|\mathbf{Z}^{L-1}, \mathbf{y}_S)\left(\prod_{l=2}^{L-1} q(\mathbf{Z}^l|\mathbf{Z}^{l-1})\right) q(\mathbf{Z}^1|\mathbf{X})d\mathbf{Z}^{1:L-1}. \qquad (15)$$

where the first term is a Gaussian: $q(\mathbf{F}_{\bar{S}}^L|\mathbf{Z}^{L-1}, \mathbf{y}_S) = \int \hat{q}(\mathbf{F}_S^L)p(\mathbf{F}_{\bar{S}}^L|\mathbf{F}_S^L, \mathbf{Z}^{L-1})d\mathbf{F}_S^L$. The conditional probability $q(\mathbf{Z}^l|\mathbf{Z}^{l-1})$ are obtained by integrating out $\mathbf{F}^l$ and also follows a Gaussian distribution. However, the latent variable $\mathbf{Z}^l$ depend non-linearly on $\mathbf{Z}^{l-1}$ through the kernel and hence the marginal can not be computed analytically. We use Monte-Carlo sampling to obtain the samples from the marginal and recursively draw the i'th sample $\hat{\mathbf{Z}}_{(i)}^l \sim q(\mathbf{Z}^l|\mathbf{Z}_{(i)}^{l-1})$ for $l = 1, 2, \ldots, L-1$ with $\hat{\mathbf{Z}}^0 = \mathbf{X}$ and use in $q(\mathbf{F}_{\bar{S}}^L|\mathbf{Z}^{L-1}, \mathbf{y}_S)$ to obtain $\mathbf{F}_{\bar{S}}^L$ samples.

$$q(\mathbf{F}_{\bar{S}}^L) = \frac{1}{T}\sum_{i=1}^{T} q(\mathbf{F}_{\bar{S}}^L|\hat{\mathbf{Z}}_{(i)}^{L-1}, \mathbf{y}_S) \qquad (16)$$

In order to facilitate gradient computation, we use the reparameterization trick to obtain the samples.

The variational parameters $\boldsymbol{\lambda}_S = \{\boldsymbol{\mu}_S^l, \boldsymbol{\Sigma}_S^l\}_{l=1}^L$ and kernel hyper-parameters are learnt by maximizing the variational lower bound (11). Computation of proposed lower bound has complexity of $\mathcal{O}(NM^2DL)$, where M is the subset size, N is number of data points, L is number of layers and $D = \max\{D^1, D^2, \ldots, D^L\}$

Prediction can be performed in a similar fashion as the computation of $q(\mathbf{F}_{\bar{S}}^L)$, where instead of the data points not in

[1] A detailed derivation of the lower bound and $q(\mathbf{F}_{\bar{S}}^L)$ is provided in the supplementary material.

the subset ($\mathbf{X}_{\bar{S}}$), test data points are used and we sample the final layer function values associated with the test samples using (15). Predictive distribution of $\mathbf{F}_*^L$ for test data points $\mathbf{X}_*$ is computed using samples of $\mathbf{Z}_*^{L-1}$ and $\mathbf{Z}_S^{L-1}$ which are in turn obtained through the reparameterization trick.

$$q(\mathbf{F}_*^L) = \frac{1}{T}\sum_{i=1}^{T} q(\mathbf{F}_*^L | \{\hat{\mathbf{Z}}_{*,(i)}, \hat{\mathbf{Z}}_{S,(i)}\}^{L-1}, \mathbf{y}_S) \qquad (17)$$

4 EXPERIMENTS

We conduct experiments to evaluate the performance of the proposed inference technique for deep Gaussian processes on various regression datasets [2]. The proposed inference technique is compared against baselines and the existing state-of-the-art inference techniques for DGPs, to demonstrate its effectiveness.

Architecture The DGP architecture is chosen to the same as that of [Salimbeni and Deisenroth, 2017]. Input layer of the DGP has D nodes, where D is dimension of input data point. In case of regression task, final layer or output layer has number of nodes set to one. Number of nodes are same accross all hidden or latent layers, with each latent or hidden layer has min(30,D) number of nodes.

Subset Selection We use the subset of training data points as the inducing inputs. We choose the subset as the collection of data points closest to the centroids obtained after K-means clustering, where K is set as the subset size (M). Subset size of 50 and 100 is used for small and medium size datasets respectively.

Model and Variational Parameters For variational distribution $q(\mathbf{F}_S^l)$, all variational mean vectors are initialised with random vectors and variational covariance matrix are initialised with identity matrix(scaled by 10^{-5} except the final layer). The hyper-parameters associated with the model are the kernel parameters and noise variance in each layer. For all the reported results for our model, both kernel variance and lengthscale parameters are initialised with the value of 0.5, and noise variance is initialised to 0.01 in the final layer and 10^{-5} in the intermediate layers.

Evalauation Metrics Negative log predictive probability (NLPP) or negative log likelihood and root mean square error (RMSE) on test data is used to report performance on regression datasets. NLPP score consider the confidence in the predictions (lower is better) and are more significant in evaluating performance of probabilistic models. While, RMSE computes the error between point predictions (mean values) and the actual observed outputs (lower is better) ignoring the variance around predictions.

[2]Code available at https://github.com/brain-iith/SOD_DGP

Training and Preprocessing Inputs and outputs in training set are scaled to zero mean unit standard deviation and the same scaling is applied in the test data set for evaluation. We consider multiple random splits of training and test data with 10% of the data as test data. We report the average RMSE and NLPP scores averaged over 5 runs. Training (lower bound maximization) is done for 20,000 iterations with batch size of 2000, using Adam optimizer initialised with learning rate of 0.01. For sampling the latent representations, we use 10 samples during training and 50 samples during prediction.

Baselines The DGP with the proposed subset of data (SoD) inference technique (SoD-DGP) is compared against a single layer sparse variational GP (SVGP) [Titsias, 2009], and state-of-the-art DGP approaches such as doubly stochastic variational inference based DGP (DSVI-DGP) [Salimbeni and Deisenroth, 2017] and stochastic gradient Hamiltonian Monte Carlo (SGHMC) based DGP (SGHMC-DGP) [Havasi et al., 2018]. All the previous approaches consider inducing inputs in each layer as learnable parameters while doing approximate inference for DGPs. The proposed approach use a subset of dataset as inducing inputs and in each layer the inducing inputs are obtained by sampling from the conditional distribution $q(\mathbf{Z}^l|\mathbf{Z}^{l-1})$. We also consider a baseline, SoD*-DGP to check the sensitivity of the subset selection strategy towards the SoD-DGP performance. SoD*-DGP uses the same SoD-DGP variational lower bound to learn the parameters but subset selection is done by randomly choosing the subset of data points instead of K-Means clustering based selection. Similarly, we consider a baseline DSVI*-DGP, where inference is done using DSVI but inducing inputs are not treated as learnable parameters but are fixed to the initial values. Here, inducing inputs for first layer is initialized based on centroids of K-means clustering, and then a PCA mapping is done as discussed in Salimbeni and Deisenroth [2017] to initialize intermediate layer inducing inputs. Unlike DSVI*-DGP, inducing input samples in SoD-DGP changes after every iteration as the conditional distribution $q(\mathbf{Z}^l|\mathbf{Z}^{l-1})$ evolves. The advantage of the proposed approach is visible from the experimental results. Model hyperparameters for the baseline methods are initialised and tuned with the values reported for these approaches [Salimbeni and Deisenroth, 2017, Havasi et al., 2018].

Regression Results We consider 7 standard small to large sized UCI regression benchmark datasets: Boston, Concrete, Energy, Winered, Protein, Naval and Year to evaluate the performance of the models. Table 1 and Fig. 4 provide the NLPP and RMSE results (mean and standard deviation) respectively for DSVI-DGP, DSVI*-DGP, SGHMC-DGP and the proposed approach SoD-DGP [3]. We consider DGP models with different number of hidden layers DGP1 (1 hid-

[3]Exact values vary slightly from the prior work because the training and test splits used are different.

Table 1: Negative log predictive probability (NLPP) score for various DGP inference techniques on UCI regression datasets. For each dataset, N represents number of training samples and D represent the input dimension. Mean NLPP scores are reported averaged over 5 runs with variance inside the brackets. Best performing model (lowest NLPP score) for each dataset is highlighted in the respective column. SoD-DGPx indicates our method. SGHMC-DGPx is the approach of Havasi et al. [2018]. DSVI is the approach of Salimbeni and Deisenroth [2017]. SoD* is same as SoD but with randomly selected subset of data points. DSVI* is same as DSVI but the inducing inputs fixed and not trained. SVGP is the approach of Titsias [2009].

Model	**Boston** N=506 D=13	**Concrete** N=1030 D=8	**Energy** N=768 D=8	**Winered** N=1599 D=22	**Protein** N=45730 D=9	**Naval** N=11934 D=26	**Year** N=515344 D=90
SoD-DGP1	2.520(0.051)	3.285(0.049)	1.927(0.213)	0.956(0.062)	2.996(0.003)	**-8.15(0.19)**	3.695(0.071)
SoD-DGP2	2.395(0.142)	**3.058(0.087)**	0.737(0.089)	**0.938(0.074)**	2.83(0.005)	-7.05(0.12)	3.595(0.003)
SoD-DGP3	**2.366(0.114)**	3.060(0.081)	0.654(0.148)	1.023(0.092)	2.753(0.006)	-7.26(0.25)	3.587(0.002)
SoD-DGP4	2.430(0.185)	3.058(0.086)	**0.568(0.106)**	0.957(0.076)	3.028(0.245)	-6.99(0.29)	3.582(0.005)
SoD*-DGP2	2.606(0.022)	3.223(0.042)	1.227(0.167)	0.979(0.062)	2.867(0.011)	-5.96(0.79)	3.639(0.087)
SoD*-DGP3	3.55(0.071)	3.284(0.015)	1.323(0.076)	1.213(0.054)	2.807(0.008)	-6.84(0.23)	3.72(0.107)
SoD*-DGP4	3.55(0.071)	3.635(0.088)	1.799(0.159)	1.213(0.054)	2.788(0.009)	-6.41(0.45)	3.696(0.108)
SGHMC-DGP2	3.217(0.442)	3.484(0.292)	3.270(5.602)	2.696(1.828)	2.789(0.034)	-5.49(0.82)	3.408(0.010)
SGHMC-DGP3	5.026(0.861)	3.384(0.224)	1.636(2.342)	3.133(1.575)	2.782(0.062)	-5.43(0.83)	**3.397(0.002)**
SGHMC-DGP4	7.736(2.440)	3.856(0.574)	2.097(3.663)	2.639(2.162)	2.743(0.025)	-5.51(0.75)	3.388(0.003)
DSVI-DGP2	2.43(0.062)	3.105(0.05)	0.761(0.119)	0.951(0.058)	2.815(0.010)	-6.97(0.06)	3.587(0.004)
DSVI-DGP3	2.427(0.059)	3.114(0.053)	0.742(0.134)	0.951(0.057)	2.755(0.004)	-6.69(0.37)	3.577(0.004)
DSVI-DGP4	2.429(0.052)	3.127(0.066)	0.732(0.131)	0.951(0.057)	**2.733(0.013)**	-5.07(1.88)	3.575(0.004)
DSVI*-DGP2	2.534(0.066)	3.185(0.026)	1.261(0.053)	0.969(0.061)	2.871(0.006)	-6.30(0.25)	3.597(0.006)
DSVI*-DGP3	2.535(0.064)	3.190(0.025)	1.270(0.061)	0.970(0.061)	2.835(0.012)	-5.32(1.33)	3.583(0.005)
DSVI*-DGP4	2.537(0.063)	3.192(0.030)	1.296(0.036)	0.969(0.06)	2.983(0.201)	-2.80(0.02)	3.581(0.005)
SVGP	2.455(0.054)	3.156(0.023)	1.282(0.056)	0.953(0.059)	2.911(0.009)	-7.42(0.18)	3.600(0.005)

den layer), DGP2 (2 hidden layers), DGP3 (3 hidden layers) and DGP4 (4 hidden layers). With respect to NLPP score, we can observe from Table 1 that the SoD-DGP approach performs better than other DGP inference approaches in all the datasets except Protein where DSVI-DGP gives the best performance. In this case also, SoD-DGP performance is very close. We find that the DSVI-DGP models perform better than DSVI*-DGP models, which does not optimize and learn the inducing inputs but fixes it to initial values. We can also observe that SoD-DGP results are in general better than SoD*-DGP, which considers random subset of data points. We find this to be more observable in deeper DGP models, while SoD*-DGP2 can provide results closer to SoD-DGP models. Thus, SoD-DGP models can be sensitive to the subset used and their performance can be further improved using a better subset selection strategy than the

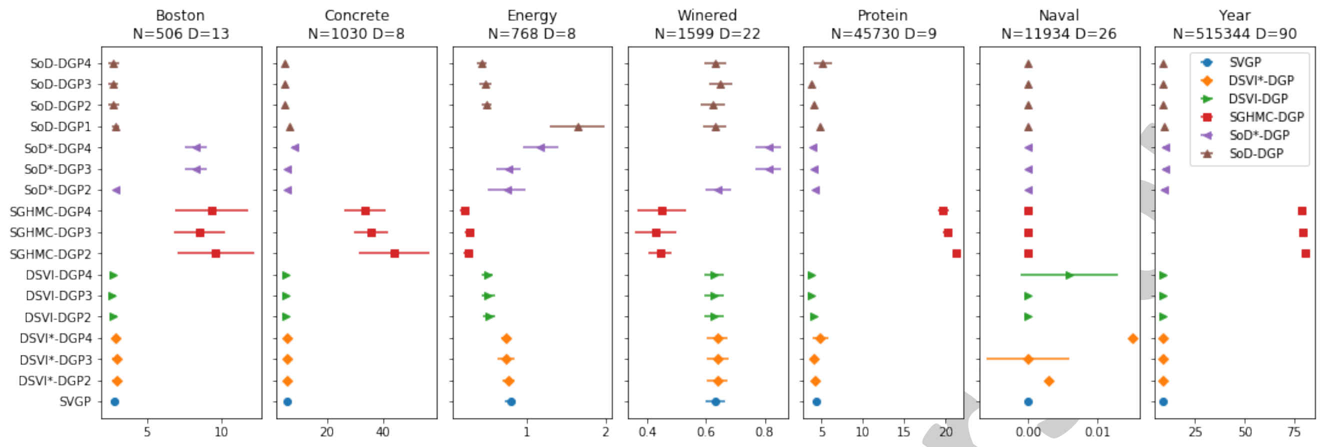


Figure 4: RMSE scores (mean along horizontal bars representing the standard deviation) for various DGP inference techniques on the UCI Regression datasets. Lower values (to the left) provide better results.

presently used naive K-Means clustering subset selection. The SVGP model gives a competitive performance on relatively 'easy' regression datasets as reported also in Salimbeni and Deisenroth [2017]. In the easy 'Naval' dataset, all the methods show very low NLPP values [4] and SVGP gives best performance while SoD-DGP gives second best result. The NLPP results also suggest that SoD-DGP provides better confidence in its predictions and might have better uncertainty modelling capabilities.

Fig. 4 provides RMSE results for various inference techniques for DGP models with lower values towards left side giving better results. We can observe that SoD-DGP models performs better or close to DSVI-DGP models and gives best performance for most of the data sets. In 2 datasets, Energy and Winered, SGHMC-DGP gives better performance. Similar to NLPP results, SoD-DGP and DSVI-DGP performs better than DSVI*-DGP. Thus, the proposed SoD-DGP model provides a computational advantage in terms of reducing the number of parameters while improving or maintaining the generalization performance.

5 CONCLUSION

We have proposed a new inference technique for deep Gaussian processes which could overcome some limitations of the existing inference techniques for DGPs. Existing inference technique require estimating large number of inducing inputs which grows with the number of layers. We propose an inference technique where inducing inputs are set to a fixed subset in training data for the first layer and the inducing inputs for subsequent layers are sampled from the conditional variational posterior. This approach reduces the number of parameters to be estimated while maintaining the generalization performance of the DGP model. This is evident from the experimental results on UCI regression datasets, where we found that SoD-DGP gives better log-likelihood values than other DGP inference techniques. The proposed approach is also amenable to sampling techniques like leverage scoring and determinantal point processes which could further improve the performance through better choice of subsets. As a future work, we will extend the proposed approach for classification problems.

[4]For Naval, NLPP is negative and RMSE is close to zero as it is a simpler problem.

Acknowledgement: MEK would like to thank a number of people who, over the years, spent time in checking the validity of (7), including Wu Lin, Heiko Strathman, Didrik Nielsen, Si Kai Lee, Anand Subramanian, Paul Cheng, and Arno Solin. AJ and PKS thank the funding and travel support from Science and Engineering Research Board (SERB), India and Japan International Co-operation Agency (JICA), Japan.

References

Ahmed Alaoui and Michael W Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems*, pages 775–783, 2015.

Thang Bui, Daniel Hernández-Lobato, Jose Hernandez-Lobato, Yingzhen Li, and Richard Turner. Deep Gaussian processes for regression using approximate expectation propagation. In *International Conference on Machine Learning*, pages 1472–1481, 2016.

Kurt Cutajar, Edwin V. Bonilla, Pietro Michiardi, and Maurizio Filippone. Random feature expansions for deep Gaussian processes. In *International Conference on Machine Learning*, volume 70, pages 884–893, 2017.

Zhenwen Dai, Andreas Damianou, Javier González, and

Neil Lawrence. Variational auto-encoded deep Gaussian processes. *International Conference on Learning Representations (ICLR)*, 2016.

Andreas Damianou. Deep Gaussian processes and variational propagation of uncertainty. *PhD Thesis, University of Sheffield*, 2015.

Andreas Damianou and Neil Lawrence. Deep Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 207–215, 2013.

Marton Havasi, José Miguel Hernández-Lobato, and Juan José Murillo-Fuentes. Inference in deep Gaussian processes using stochastic gradient Hamiltonian monte carlo. In *Advances in Neural Information Processing Systems*, pages 7517–7527, 2018.

James Hensman and Neil D Lawrence. Nested variational compression in deep Gaussian processes. *arXiv preprint arXiv:1412.1370*, 2014.

James Hensman, Nicolo Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI2013)*, 2013.

Tarun Kathuria, Amit Deshpande, and Pushmeet Kohli. Batched Gaussian process bandit optimization via determinantal point processes. In *Advances in Neural Information Processing Systems*, pages 4206–4214, 2016.

Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems 30*, pages 4588–4599. 2017.

S. Shevade and S. Sundararajan. Validation-based sparse Gaussian process classifier design. *Neural Computation*, 21(7):2082–2103, July 2009.

Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 567–574, 2009.

Haibin Yu, Yizhou Chen, Bryan Kian Hsiang Low, Patrick Jaillet, and Zhongxiang Dai. Implicit posterior variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 14475–14486, 2019.