# Deep Gaussian Processes with Importance-Weighted Variational Inference

**Hugh Salimbeni** [1 2]   **Vincent Dutordoir** [2]   **James Hensman** [2]   **Marc Peter Deisenroth** [1 2]

## Abstract

Deep Gaussian processes (DGPs) can model complex marginal densities as well as complex mappings. Non-Gaussian marginals are essential for modelling real-world data, and can be generated from the DGP by incorporating uncorrelated variables to the model. Previous work on DGP models has introduced noise additively and used variational inference with a combination of sparse Gaussian processes and mean-field Gaussians for the approximate posterior. Additive noise attenuates the signal, and the Gaussian form of variational distribution may lead to an inaccurate posterior. We instead incorporate noisy variables as latent covariates, and propose a novel importance-weighted objective, which leverages analytic results and provides a mechanism to trade off computation for improved accuracy. Our results demonstrate that the importance-weighted objective works well in practice and consistently outperforms classical variational inference, especially for deeper models.

## 1. Introduction

Gaussian processes are powerful and popular models with widespread use, but the joint Gaussian assumption of the latent function values can be limiting in many situations. This can be for at least two reasons: firstly, not all prior knowledge is possible to express solely in terms of mean and covariance, and secondly Gaussian marginals are not sufficient for many applications. The deep Gaussian process (DGP) (Damianou and Lawrence, 2013) can potentially overcome both these limitations.

We consider the very general problem of conditional density estimation. A Gaussian process model with a Gaussian likelihood is a poor model unless the true data marginals are Gaussian. Even if the marginals are Gaussian, a Gaus-
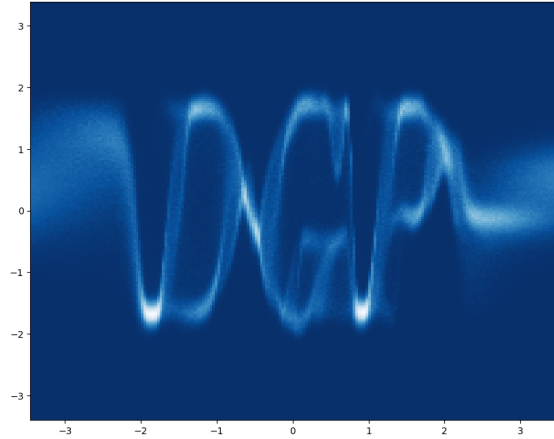


Figure 1: Posterior density from a two-layer model, illustrating non-Gaussian marginals and non-smooth input dependence.

sian process can still be a poor model if the mapping is not well-captured by a known mean and covariance. For example, changes of lengthscale at unknown locations cannot be captured with a fixed covariance function (Neil, 1998).

As an example, consider a 1D density estimation task formed from the letters '*DGP*', where the inputs are the horizontal coordinates and the outputs are the vertical coordinates of points uniformly in the glyphs. The marginals are multimodal and piece-wise constant, and also change non-smoothly in some regions of the space (for example changing discontinuously from a bi- to tri-modal density mid-way through the 'G'). Figure 1 shows the posterior of a two-layer DGP with this data.

In this work, we develop the DGP model by revisiting the original construction of Damianou and Lawrence (2013), which includes variables that are a priori independent for each data point. Independent variables are important for modelling non-Gaussian marginals, as otherwise function values with the same input will be perfectly correlated. Differently from Damianou and Lawrence (2013), we introduce the independent variables to the model as latent covariates (that is, as additional inputs to the GP) (Wang and Neal, 2012) rather than additively as process noise. As we have a clean separation between variables that are correlated and

---

[1] Imperial College London [2] PROWLER.io. Correspondence to: Hugh Salimbeni <hrs13@ic.ac.uk>.

variables that are independent, we can apply appropriate inference for each. For the correlated variables we use a sparse variational GP to represent the posterior (Titsias, 2009; Matthews et al., 2016). For the uncorrelated variables we use a mean-field approach. We first derive a straightforward combination of the stochastic variational approach of Salimbeni and Deisenroth (2017) with a mean-field Gaussian approximation for the latent variables. We then propose a novel importance-weighted scheme that improves upon the Gaussian approach and trades additional computation for accuracy while retaining analytic results for the GP parts.

Our results show that the deep Gaussian process with latent variables is an effective model for real data. We investigate a large number of datasets and demonstrate that highly non-Gaussian marginals occur in practice, and that they are not well modelled by the noise-free approach of Salimbeni and Deisenroth (2017). We also show that our importance-weighted scheme is always an improvement over variational inference, especially for the deeper models.

## 2. Model

Our DGP model is built from two components (or *layers*) that can be freely combined to create a family of models. The two types of layers have orthogonal uses: one is for modelling *epistemic* uncertainty (also known as *model* or *reducible* uncertainty, when the output depends deterministically on the input, but there is uncertainty due to lack of observations) and the other is for modelling *aleatoric* uncertainty (also known as *irreducible* uncertainty, when the output is inherently random). Both layers are Gaussian processes, but we use the term *latent variable* when the process has the white noise covariance, and we use *Gaussian process* when the covariance has no noise component. Each layer takes an input and returns an output. The model is constructed by applying the layers sequentially to the input $x_n$ to get a density over an output $y_n$.

### 2.1. Gaussian process (GP) layer

The Gaussian process layer defines a set of continuously indexed random variables, which are a priori jointly Gaussian, with mean and covariance that depend on the indices. We write $f$ for the full set of variables (or 'function') and $f(x)$ the particular variable with index $x$. The notation $f \sim \mathcal{GP}(\mu, k)$ states that for any finite set $\{x_i\}$, the variables $\{f(x_i)\}$ are jointly Gaussian, with $\mathbb{E}\left((f(x_i)) = \mu(x_i)\right)$ and $\mathrm{cov}\left(f(x_i), f(x_j)\right) = k(x_i, x_j)$, where $\mu$ is a mean function and $k$ is a positive semi-definite covariance function. In this work, we always use 'noise free' kernels, which satisfy $k(x, x) = \lim_{x' \to x} k(x, x')$.

The GP prior is defined for all inputs, not just the ones associated with data. When we notate the GP function in
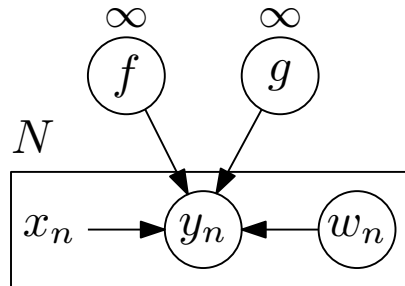


Figure 2: Graphical model for LV-GP-GP

a graphical model (see Figure 2), all the variables appear together as $f$. We include an infinity symbol in the graphical model to indicate that the GP node represents an infinite collection of random variables, one for every possible input.

### 2.2. Latent-variable (LV) layer

The latent-variable layer introduces variables to the model that are independent for each data point. As we wish to interpret these variables as unobserved covariates we introduce them through concatenation rather than through addition, as in previous work (see, for example Dai et al., 2015; Bui et al., 2016; Damianou and Lawrence, 2013). We use the notation $[x_n, w_n]$ to denote the concatenation of $x_n$ with $w_n$. Throughout this work each component of $w_n$ will be distributed as $\mathcal{N}(0, 1)$. For input $x_n$, the output of the latent variable layer is $[x_n, w_n]$.

### 2.3. Example model: LV-GP-GP

For the purpose of demonstrating the details of inference in the next section, we focus on a particular model. The model has an initial layer of latent variables followed by two Gaussian process layers. We refer to this model as 'LV-GP-GP'. The graphical model indicating the a priori statistical relationship between the variables is shown in Figure 2. Writing $y$ and $w$ for $\{y_n\}$ and $\{w_n\}$, the likelihood of this model is $p(y|f, g, w) = \prod_n p(y_n|f, g, w_n)$, with

$$p(y_n|f, g, w_n) = \mathcal{N}(y_n|f(g([x_n, w_n])), \sigma^2). \quad (1)$$

The priors are

$$w_n \sim \mathcal{N}(0, 1), \; f \sim \mathcal{GP}(\mu_1, k_1), \; g \sim \mathcal{GP}(\mu_2, k_2).$$

The model is appropriate for continuously valued data, and can model heavy and asymmetric tails, as well as multi-modality, as we later demonstrate. We discuss choices for the mean and covariance functions in section 2.6.

### 2.4. Model variants

The LV-GP-GP model can be extended by adding more layers, and by adding more latent variables. For example,

we could insert an additional GP layer,

$$p(y|f, g, h, w) = \prod_n \mathcal{N}\left(y_n | h\left(f\left(g\left([x_n, w_n]\right)\right)\right), \sigma^2\right),$$

where $h \sim \mathcal{GP}(\mu_3, k_3)$. We refer to this model as LV-GP-GP-GP. Alternatively we could place the latent variables between the GPs, and have instead $f([g(x_n), w_n])$ for the conditional mean. We refer to this model as GP-LV-GP. Other models are analogously defined. For example, GP-GP is as above but with no latent variable. We refer to models without latent variables as 'noise-free'.

Prior samples from models with one and two GP layers are shown in Figure 3 to illustrate their properties. All the GP layers have 1D outputs and use the RBF kernel with unit lengthscale. Further illustrations with additional samples and additional models are shown in the supplementary material, Figures 6-9. For the latent variable models, the number of GPs above the latent variable layer determines the complexity of the marginals, but the number of GPs in total determines the complexity of the functional mapping.

### 2.5. Multiple outputs

All variables in our model (including the model inputs and outputs) may be vector valued. The 'multiple output GP' is a GP with a covariance function defined between outputs $k_{dd'}(x, x')$, where $d$ indexes output dimension and $x$ indexes the input. In this notation, the independent output approach of Damianou and Lawrence (2013) can be written as $k_{dd'}(x, x') = \delta_{dd'} k(x, x')$. For all our models, we consider a linear model of covariance between outputs: $k_{dd'}(x, x') = \sum_e P_{de} k_e(x, x') P_{ed'}$ (Alvarez et al., 2012). For $k_e(x, x') = k(x, x')$ this is equivalent to assuming a Kronecker structured covariance between the outputs stacked into a single vector (i.e. with covariance $(P^\top P) \otimes K$), and is also equivalent to multiplying independent GPs stacked into a vector by the matrix $P$. Note that we are free to take $e < d$. When there are multiple observed outputs we assume the likelihood factorizes over the outputs, and in this work we assume independent GPs in the final layer. A priori correlations between outputs are straightforward to include using the linear model, following Dai et al. (2017); Dutordoir et al. (2018).

### 2.6. Mean and covariance functions

The DGP model suffers from a pathology if used with zero mean GPs at each layer (Duvenaud et al., 2014; Dunlop et al., 2018). To remedy this problem, Duvenaud et al. (2014) propose concatenating the inputs with the outputs at each GP layer, whereas Salimbeni and Deisenroth (2017) use a linear mean function to address this issue. We follow the latter approach. The inference we present in the next section is agnostic to covariance function, so we are free to

make any choice. In our experiments, we use an RBF kernel for each layer, sharing the kernel over the different outputs.

## 3. Inference in the LV-GP-GP model

In this section, we present two approaches for approximate inference: the first with variational inference and a second with importance-weighted variational inference. Both schemes can optionally amortize the optimization of the local variational parameters (often referred to as 'auto-encoding'), are scalable through data sub-sampling, and can exploit natural gradients of the variational parameters for the final layer. While the variational approach is a straightforward extension of the doubly-stochastic method by Salimbeni and Deisenroth (2017), the importance-weighted approach is more subtle and requires a careful choice of variational distribution to retain the analytic results for the final layer.

### 3.1. Variational inference

Variational inference seeks an approximate posterior that is close to the true posterior in terms of KL divergence. The posterior is typically restricted to some tractable family of distributions, and an optimization problem is formed by minimizing the KL divergence from an approximate posterior to the true posterior. Equivalently, the same objective can be obtained by applying Jensen's inequality to an importance-weighted expression for the marginal likelihood (Domke and Sheldon, 2018). For an approximate posterior, we follow Damianou and Lawrence (2013) and use a mean-field Gaussian distribution for the latent variables $q(w) = \prod_n q(w_n)$ with $q(w_n) = \mathcal{N}(a_n, b_n)$, and independent processes for the functions. The posterior density[1] then has the same structure as the prior: $q(w, f, g) = q(w)q(f)q(g)$. We begin by writing the (exact) marginal likelihood as

$$p(y) = \mathbb{E}_{f,g,w}\left[p(y|f, g, w) \frac{p(f)p(g)p(w)}{q(f)q(g)q(w)}\right], \quad (2)$$

where the expectations are taken with respect to the variational distributions. Applying Jensen's inequality to the logarithm of both sides of (2), we obtain

$$\log p(y) \geq \sum_n \left(A_n - \text{KL}_{w_n}\right) - \text{KL}_f - \text{KL}_g, \quad (3)$$

where we have used the short-hand $\text{KL}_h$ for $\text{KL}(q(h)||p(h))$, and $A_n$ is given by

$$A_n = \mathbb{E}_{f,g,w_n} \log p(y_n|f, g, w_n). \quad (4)$$

By considering a variational distribution over the entire function for $f$ and $g$ we avoid the difficulty of representing the

---

[1]We abuse notation and write a process as if it has a density. The derivation can be made rigorous as the densities only appear as expectations of ratios. See Matthews et al. (2016) for details.
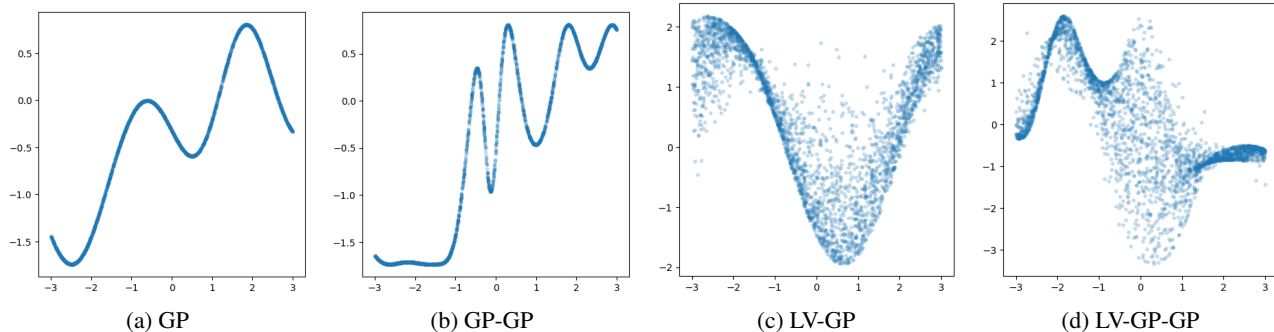
Figure 3: A single sample from the prior for 4 illustrative models. The GP and GP-GP model have no latent variables. Their samples appear as deterministic functions. See the supplementary material for more examples.

indeterminately indexed inner-layer variables. We require only that $\text{KL}_f$ and $\text{KL}_g$ are finite, which is possible if we construct $q(f)$ and $q(g)$ as measures with respect to their priors. We follow Hensman et al. (2013) and form these posteriors by conditioning the prior on some finite set of 'inducing' points with a parameterized Gaussian distribution. We defer the details to the supplementary material (Section B) as the results are widely known (see, for example Matthews et al., 2016; Cheng and Boots, 2016; Dutordoir et al., 2018). The priors and variational posteriors over $f$ and $g$ are independent (the coupling is only in the likelihood), so the results from the single layer GP apply to each layer.

To evaluate $A_n$ exactly is intractable except in the single-layer case (and then only with a certain kernel; see Titsias and Lawrence (2010) for details), so we rely on a Monte Carlo estimate. To obtain unbiased samples we first reparameterize the expectations over $w_n$ and $g$, where $w_n = a_n + \epsilon_1 \sqrt{b_n}$ and $g([x_n, w_n]) = \mu_2([x_n, w_n]) + \epsilon_2 \sqrt{k_2([x_n, w_n], [x_n, w_n])}$ where $\epsilon_1, \epsilon_2 \sim \mathcal{N}(0, 1)$. We then obtain an estimate of $A_n$ by sampling from $\epsilon_1, \epsilon_2$, which is the 'reparameterization trick' popularized by Rezende et al. (2014); Kingma and Welling (2014). After these two expectations have been approximated with a Monte Carlo estimate we can take the expectation over $f$ analytically as the likelihood is Gaussian.

### 3.2. Importance-weighted variational inference

Jensen's inequality is tighter if the quantity inside the expectation is concentrated about its mean (Domke and Sheldon, 2018). To decrease the variance inside (2) while preserving the value, we can replace the $w$ term with a sample average of $K$ terms,

$$p(y) = \mathop{\mathbb{E}}_{f,g,w} \frac{1}{K} \sum_{k=1}^{K} p(y|f, g, w^{(k)}) \frac{p(w^{(k)})}{q(w^{(k)})} \frac{p(f)p(g)}{q(f)q(g)} . \quad (5)$$

The expression inside the expectation in (5) is known as the

importance sampled estimate (with respect to $w$), motivating the term 'importance-weighted variational inference' when we take the logarithm of both sides and form a lower bound using Jensen's inequality. Applying Jensen's inequality to (5), we have the lower bound

$$\log p(y) \geq \sum_{n=1}^{N} B_n - \text{KL}_f - \text{KL}_g , \quad (6)$$

where the data-fit term $B_n$ is given by

$$B_n = \mathop{\mathbb{E}}_{f,g,w_n} \log \frac{1}{K} \sum_{k} p(y_n|f, g, w_n^{(k)}) \frac{p(w_n^{(k)})}{q(w_n^{(k)})} . \quad (7)$$

This approach provides a strictly tighter bound (Burda et al., 2016; Domke and Sheldon, 2018), but the expression for $B_n$ is less convenient to estimate than $A_n$ as we cannot apply the analytic result for the $f$ expectation due to the non-linearity of the logarithm. We must therefore resort to sampling for the expectation over $f$ as well as $g$ and $w_n$, incurring potentially high variance. This seems like an unacceptable price to pay as we have not gained any additional flexibility over $f$. In the next section, we show how we can retain the analytic expectation for $f$ while exploiting importance-weighted sampling for $w$.

### 3.3. Analytic final layer

The expression in (7) does not exploit the conjugacy of the Gaussian likelihood with the final layer. In this section, we present a novel two-stage approach to obtain a bound that has all the analytic properties of the variational bound (3), but with improved accuracy. Our aim is to obtain a modified version of (7) but with the $f$ expectation taken over the logarithm of the likelihood, since this expectation is tractable. We will show that we can find a bound that replaces the term $p(y_n|f, g, w_n)$ in (7) with $\exp \mathbb{E}_f \log p(y_n|f, g, w_n)$, which we can compute analytically. It is worth noting that since

the likelihood is Gaussian we could analytically integrate out $f$ to obtain $p(y|g, w)$, though doing so precludes the use of minibatches and incurs $\mathcal{O}(N^3)$ complexity. It is not surprising, then, that it is possible to 'collapse' the bound over $f$ approximately. Our approach is inspired by the partially collapsed approaches in Hensman et al. (2012). We begin by applying Jensen's inequality to the $f$ expectation in (2):

$$\log p(y|g, w) \geq \sum_n L_n(g, w_n) - \text{KL}_f,$$

where the data term is given by

$$L_n(g, w_n) = \mathbb{E}_f \log p(y_n|f, g, w_n). \qquad (8)$$

The expression for $L_n(g, w_n)$ is available in closed form as the conditional likelihood is Gaussian (see, for example, Titsias, 2009) . Applying the exponential function to both sides of (8) gives the bound

$$p(y|g, w) \geq \exp \left[ \sum_n L_n(g, w_n) - \text{KL}_f \right] . \qquad (9)$$

Returning again to (2), the exact marginal likelihood can be expressed equivalently with $f$ marginalized as

$$p(y) = \mathop{\mathbb{E}}_{g,w} p(y|g, w) \frac{p(w)p(g)}{q(w)q(g)} . \qquad (10)$$

We can now use the bound on $p(y|g, w)$ from (9) and substitute into (10) to obtain

$$p(y) \geq \mathop{\mathbb{E}}_{g,w} \exp \left[ \sum_n L_n(g, w_n) - \text{KL}_f \right] \frac{p(w)p(g)}{q(w)q(g)} .$$

Next we use Jensen's inequality for the $g$ expectation. After some rearranging the bound is given by

$$\log p(y) \geq \sum_n \mathbb{E}_g \underbrace{\log \mathbb{E}_w \frac{e^{L_n(g, w_n)} p(w_n)}{q(w_n)}}_{T_n(g)} - \text{KL}_f - \text{KL}_g .$$

To bound $T_n(g)$, we first reduce the variance of the quantity inside the expectation using the sample average as before to tighten the subsequent use of Jensen's inequality. For any $K$, $T_n(g)$ is (exactly) equal to

$$T_n(g) = \log \mathbb{E}_{w_n} \frac{1}{K} \sum_k \frac{e^{L_n(g, w_n^{(k)})} p(w_n^{(k)})}{q(w_n^{(k)})} ,$$

where $w_n^{(k)}$ are independent samples from $q(w_n)$. We now make a final use of Jensen for the $w_n$ expectation, and the final objective is then given by

$$\sum_n \mathop{\mathbb{E}}_{g,w_n} \log \frac{1}{K} \sum_k \frac{e^{L_n(g, w_n^{(k)})} p(w_n^{(k)})}{q(w_n^{(k)})} - \text{KL}_f - \text{KL}_g .$$
$$(11)$$

This bound can be evaluated using Monte Carlo sampling for $g$ and $w_n$, both with the reparameterization described in Section 3.1.

The $K$ terms inside the sum in (11) must be sampled with a single draw from $q(g)$, and not independently. This is not an insurmountable problem, however, as we can draw $K$ samples using the full covariance and reparameterize using the Cholesky factor at $\mathcal{O}(K^3)$ cost. Note that the decomposition over the $N$ data points is a consequence of our choice of proposal distributions and the factorization of the likelihood, so we do not incur $\mathcal{O}(N^3)$ complexity and can sample each term in the sum over $N$ independently.

### 3.4. The posterior over the latent variables

Variational inference simultaneously finds a lower bound and an approximate posterior. The importance weighted approach does also minimize the KL divergence posterior, but the posterior is an implicit one over an augmented space. Samples from this posterior are obtained by first sampling $w_n^k \sim q(w_n)$ and then selecting one of the $w_n^k$ with probabilities proportional to $\frac{e^{L_n(g, w_n^{(k)})} p(w_n^{(k)})}{q(w_n^{(k)})}$. We refer to Domke and Sheldon (2018) for details. In this work, we are not concerned with the posterior over the latent variables themselves, but rather with prediction at test points where we sample from the prior.

### 3.5. Further inference details

We can amortize the optimization of the $a_n, b_n$ parameters by making them parameterized functions of $(x_n, y_n)$. As the bound is a sum over the data we can use data subsampling. We can also use natural gradients for the variational parameters of the final layer, following Salimbeni et al. (2018); Dutordoir et al. (2018). Our bound is modular in both the GP and LV layers so that it extends straightforwardly to the other model variants.

## 4. Results

We use a density estimation task to establish the utility of the DGP models with and without latent variables. For the latent variable models we also compare our importance-weighted (IW) inference (11) against the variational inference (VI) approach (3). Our central interest is how well the models balance the complexity of the marginals with the complexity of the mapping from inputs to outputs.

### 4.1. 1D example

To illustrate the inductive bias of our models we show a 1D example with a conditional density that is non-Gaussian, heteroscedastic and changes discontinuously in some regions of the space. We form a dataset from the letters '*DGP*'
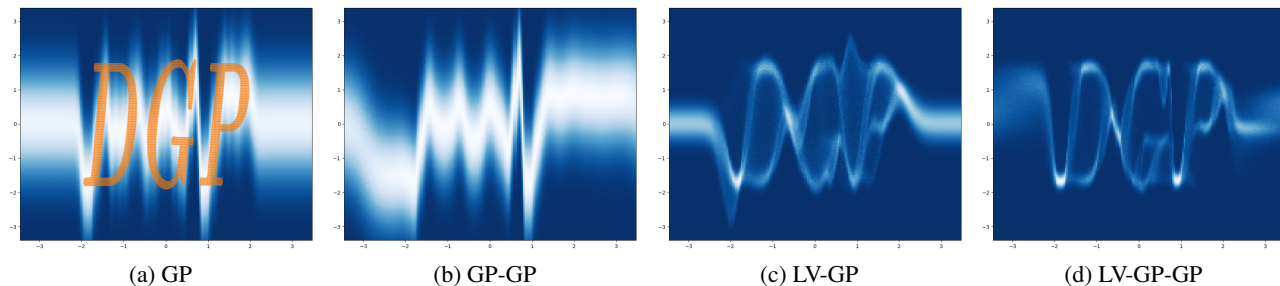
Figure 4: Posteriors for 1D synthetic data. The models without latent variables cannot capture the bimodality or heteroscedasticity. See Figure 1 for the LV-GP-GP-GP model.

by taking a fine grid of points inside the glyphs of a cropped image and using the horizontal coordinates as the input and the vertical coordinates as the output (73, 419 points in total). Posteriors are shown in Figure 4 for four models (the same models as in Figure 3), with the data plotted in the first figure. The GP model can obviously not model this data, but neither can the GP-GP as the approximate posterior has no way to separate out the points with the same input but different outputs. The LV-GP model can fit the data to some extent, and the LV-GP-GP model more closely captures the shapes of the letters.

We see that the LV-GP has a tendency to smoother densities than LV-GP-GP, and also a smoother response as a function of the input. Between the 'G' and 'P' the LV-GP-GP model extrapolates with high confidence, whereas the LV-GP model maintains a bi-modal distribution connecting the two letters. Posteriors from further models are in the supplementary material.

## 4.2. UCI datasets

We use 41 publicly available datasets[2] with 1D targets. The datasets range in size from 23 points to 2, 049, 280. In each case we reserve 10% of the data for evaluating a test log-likelihood, repeating the experiment five times with different splits. We use five samples for the importance-weighted models, 128 inducing points, and five GP outputs for the inner layers. Hyperparameters and initializations are the same for all models and datasets and are fully detailed in the supplementary material. Results for test log-likelihood are reported in Table 1 for the GP models. Table 2 in the supplementary material shows additional results for conditional VAE models (Sohn et al., 2015), and results from deep Gaussian process models using stochastic gradient Hamiltonian Monte Carlo (Havasi et al., 2018). To assess the non-Gaussianity of the predictive distribution we compute

the Shapiro–Wilk test statistic on the test point marginals. The test statistics are shown in 3 in the supplementary material. Full code to reproduce our results is available online [3]. We draw five conclusions from the results.

**1) Latent variables improve performance**. For a given depth of Gaussian process mappings (for example GP-GP compared to LV-GP-GP) the latent variable model generally has equal or better performance, and often considerably better. This is true both using IW and VI.

**2) IW outperforms VI**. The difference is more pronounced on the deeper models, which is to be expected as the deeper models are likely to have more complicated posteriors, so the VI approximation is likely to be more severe. There are also a few datasets ('keggundirected', 'servo' and 'automgp') where the VI latent variable model has very poor performance, but the IW approach performs well. This suggests that the IW approach might be less prone to local optima compared with VI.

**3) Some datasets require latent variables for any improvement over the single-layer model**. For several datasets (for example 'forest' and 'power') the inclusion of latent variables makes a very pronounced difference, whereas depth alone cannot improve over the single layer model. These datasets are highlighted in **blue**[*]. For all these datasets the marginals for the latent variable models are non-Gaussian (see Table 3 in the supplementary material). Conversely, for some datasets (e.g., 'kin40k' and 'sml') we observe much greater benefit from the deep noise free models than the single layer model with latent variables. These datasets are highlighted in **orange**[†].

**4) Some datasets benefit from both depth and latent variables**. For the 'bike' and 'keggdirected' data, for example, we see that both the LV-GP and GP-GP model improve over the single layer model, suggesting that complex input dependence and non-Gaussian marginals are beneficial for

---

[2]The full datasets with the splits and pre-processing can be found at `github.com/hughsalimbeni/bayesian_benchmarks`.

[3]`https://github.com/hughsalimbeni/DGPs_with_IWVI`

Table 1: Average test log-likelihood results for five splits (standard error). **Bold** font indicates overlapping error bars with the best performing method. *Italic* indicates a significantly non-Gaussian posterior, computed by the Shapiro–Wilk test (see Table 3). Colours indicate datasets demonstrating prominent examples of complex **marginals**[*], **mappings**[†], or **both**[‡].

| Dataset | N | D | GP VI | GP-GP VI | GP-GP-GP VI | LV-GP VI | LV-GP IWVI | LV-GP-GP VI | LV-GP-GP IWVI | LV-GP-GP-GP VI | LV-GP-GP-GP IWVI |
|---|---|---|---|---|---|---|---|---|---|---|---|
| challenger | 23 | 4 | -3.22 (0.07) | -2.04 (0.02) | -2.10 (0.03) | ***-0.67 (0.01)*** | ***-0.68 (0.00)*** | *-9.99 (3.60)* | *-9.47 (2.19)* | -1.21 (0.01) | -2.45 (0.04) |
| fertility | 100 | 9 | -2.51 (0.13) | -1.43 (0.01) | -1.40 (0.01) | -1.41 (0.01) | -1.40 (0.01) | -1.42 (0.01) | -1.69 (0.01) | **-1.32 (0.00)** | **-1.31 (0.01)** |
| concreteslump | 103 | 7 | 1.91 (0.01) | 1.55 (0.00) | 1.45 (0.00) | 1.93 (0.00) | **1.94 (0.00)** | 1.53 (0.00) | 1.75 (0.00) | 1.43 (0.00) | 1.62 (0.00) |
| autos | 159 | 25 | -0.36 (0.00) | -0.14 (0.01) | -0.14 (0.03) | -0.36 (0.01) | -0.35 (0.00) | **-0.06 (0.04)** | -0.34 (0.01) | -0.25 (0.05) | -0.33 (0.02) |
| servo | 167 | 4 | -0.17 (0.00) | -0.19 (0.01) | -0.17 (0.01) | -0.07 (0.01) | -0.08 (0.01) | -0.19 (0.01) | -0.07 (0.01) | -0.25 (0.01) | **0.03 (0.03)** |
| breastcancer | 194 | 33 | -1.32 (0.01) | -1.36 (0.01) | -1.34 (0.00) | **-1.31 (0.00)** | **-1.31 (0.00)** | -1.43 (0.01) | -1.59 (0.10) | -1.38 (0.01) | -1.80 (0.19) |
| machine | 209 | 7 | -0.70 (0.01) | -0.65 (0.02) | -0.61 (0.01) | -0.75 (0.02) | -0.71 (0.02) | -0.63 (0.01) | **-0.51 (0.01)** | -0.59 (0.01) | **-0.52 (0.03)** |
| yacht | 308 | 6 | 1.32 (0.02) | 1.84 (0.01) | **2.00 (0.06)** | 1.64 (0.00) | 1.65 (0.00) | 1.69 (0.09) | 1.71 (0.01) | 1.77 (0.07) | **2.05 (0.02)** |
| autompg | 392 | 7 | -0.44 (0.01) | -0.57 (0.02) | -0.48 (0.03) | -0.32 (0.01) | -0.33 (0.01) | -0.38 (0.04) | **-0.24 (0.01)** | -0.65 (0.11) | -0.28 (0.03) |
| boston | 506 | 13 | 0.02 (0.00) | 0.07 (0.00) | 0.03 (0.00) | -0.04 (0.00) | -0.07 (0.00) | 0.06 (0.00) | 0.14 (0.00) | -0.04 (0.01) | **0.17 (0.00)** |
| **forest**[*] | 517 | 12 | -1.38 (0.00) | -1.37 (0.00) | -1.37 (0.00) | *-0.99 (0.00)* | *-1.00 (0.00)* | *-0.91 (0.00)* | *-1.02 (0.02)* | ***-0.92 (0.01)*** | ***-0.91 (0.01)*** |
| stock | 536 | 11 | -0.22 (0.00) | -0.29 (0.00) | -0.26 (0.00) | -0.22 (0.00) | -0.22 (0.00) | -0.29 (0.00) | **-0.19 (0.00)** | -0.23 (0.01) | -0.36 (0.02) |
| pendulum | 630 | 9 | -0.14 (0.00) | 0.22 (0.01) | 0.12 (0.08) | -0.14 (0.00) | -0.14 (0.00) | 0.22 (0.01) | **0.33 (0.08)** | **0.25 (0.00)** | 0.20 (0.03) |
| energy | 768 | 8 | 1.71 (0.00) | 1.85 (0.00) | 2.07 (0.01) | 1.86 (0.01) | 1.92 (0.01) | 2.00 (0.04) | 1.96 (0.01) | 2.07 (0.01) | **2.28 (0.01)** |
| concrete | 1030 | 8 | -0.43 (0.00) | -0.45 (0.00) | -0.35 (0.00) | -0.32 (0.00) | -0.32 (0.00) | -0.35 (0.00) | -0.21 (0.00) | -0.22 (0.01) | **-0.12 (0.00)** |
| **solar**[*] | 1066 | 10 | -1.75 (0.08) | -1.21 (0.02) | -1.20 (0.03) | *0.04 (0.07)* | *0.07 (0.01)* | *0.54 (0.01)* | *0.22 (0.01)* | ***0.54 (0.01)*** | *0.20 (0.02)* |
| airfoil | 1503 | 5 | -0.79 (0.05) | 0.08 (0.02) | 0.14 (0.03) | -0.44 (0.03) | -0.36 (0.01) | 0.07 (0.00) | 0.30 (0.00) | -0.02 (0.03) | **0.34 (0.01)** |
| winered | 1599 | 11 | -1.09 (0.00) | -1.11 (0.00) | -1.08 (0.00) | -1.07 (0.00) | -1.06 (0.00) | -1.10 (0.00) | *-0.84 (0.01)* | *-1.06 (0.03)* | *-1.31 (0.40)* |
| gas | 2565 | 128 | 1.07 (0.00) | **1.60 (0.05)** | **1.69 (0.05)** | **1.69 (0.08)** | 1.56 (0.06) | 0.70 (0.10) | **1.57 (0.16)** | 1.30 (0.14) | **1.61 (0.12)** |
| skillcraft | 3338 | 19 | -0.94 (0.00) | -0.94 (0.00) | -0.94 (0.00) | **-0.91 (0.00)** | **-0.91 (0.00)** | -0.92 (0.00) | -0.93 (0.00) | -0.92 (0.00) | -0.94 (0.00) |
| **sml**[†] | 4137 | 26 | 1.53 (0.00) | 1.72 (0.01) | 1.83 (0.01) | 1.52 (0.00) | 1.52 (0.00) | 1.79 (0.00) | 1.91 (0.00) | **1.92 (0.01)** | **1.97 (0.05)** |
| winewhite | 4898 | 11 | -1.14 (0.00) | -1.14 (0.00) | -1.14 (0.00) | -1.13 (0.00) | -1.13 (0.00) | -1.13 (0.00) | **-1.10 (0.00)** | -1.13 (0.00) | **-1.09 (0.00)** |
| parkinsons | 5875 | 20 | 1.99 (0.00) | 2.61 (0.01) | 2.75 (0.02) | 1.79 (0.02) | 1.82 (0.02) | 2.36 (0.02) | 2.76 (0.02) | 2.71 (0.07) | **3.12 (0.05)** |
| **kin8nm**[†] | 8192 | 8 | -0.29 (0.00) | -0.01 (0.00) | -0.00 (0.00) | -0.29 (0.00) | -0.29 (0.00) | -0.02 (0.00) | -0.00 (0.00) | -0.00 (0.00) | **0.03 (0.00)** |
| pumadyn32nm | 8192 | 32 | 0.08 (0.00) | **0.11 (0.01)** | **0.11 (0.00)** | -1.44 (0.00) | -1.44 (0.00) | 0.10 (0.01) | -0.62 (0.25) | **0.11 (0.00)** | **0.11 (0.00)** |
| **power**[*] | 9568 | 4 | -0.65 (0.04) | -0.75 (0.03) | -0.80 (0.02) | *-0.39 (0.04)* | *-0.23 (0.02)* | *-0.36 (0.04)* | *-0.28 (0.05)* | *-0.25 (0.06)* | ***-0.11 (0.06)*** |
| naval | 11934 | 14 | **4.52 (0.02)** | 4.43 (0.03) | 4.35 (0.03) | *4.19 (0.01)* | *4.24 (0.01)* | 4.36 (0.01) | **4.52 (0.02)** | 4.27 (0.02) | 4.41 (0.02) |
| **pol**[‡] | 15000 | 26 | 0.48 (0.00) | 1.51 (0.01) | 1.45 (0.01) | *0.37 (0.08)* | *-0.50 (0.00)* | *2.34 (0.02)* | *2.63 (0.01)* | 1.47 (0.01) | **2.72 (0.10)** |
| elevators | 16599 | 18 | -0.44 (0.00) | -0.41 (0.01) | -0.40 (0.01) | -0.37 (0.00) | -0.36 (0.00) | -0.29 (0.00) | **-0.27 (0.00)** | -0.28 (0.00) | **-0.27 (0.00)** |
| **bike**[‡] | 17379 | 17 | 0.82 (0.01) | 3.49 (0.01) | 3.68 (0.01) | *2.48 (0.01)* | *2.66 (0.01)* | 3.48 (0.01) | 3.75 (0.01) | 3.72 (0.01) | **3.95 (0.01)** |
| **kin40k**[†] | 40000 | 8 | 0.02 (0.00) | 0.84 (0.00) | 1.17 (0.00) | 0.04 (0.00) | 0.05 (0.00) | 0.87 (0.01) | 0.93 (0.00) | 1.15 (0.00) | **1.27 (0.00)** |
| protein | 45730 | 9 | -1.06 (0.00) | -0.98 (0.00) | -0.95 (0.00) | *-0.85 (0.00)* | *-0.80 (0.00)* | *-0.70 (0.00)* | *-0.61 (0.00)* | *-0.68 (0.00)* | ***-0.57 (0.00)*** |
| tamielectric | 45781 | 3 | -1.43 (0.00) | -1.43 (0.00) | -1.43 (0.00) | **-1.31 (0.00)** | -1.31 (0.00) | **-1.31 (0.00)** | -1.31 (0.00) | -1.31 (0.00) | -1.31 (0.00) |
| **keggdirected**[‡] | 48827 | 20 | 0.16 (0.01) | 0.21 (0.01) | 0.26 (0.02) | *1.24 (0.06)* | *2.03 (0.01)* | *1.84 (0.05)* | *2.23 (0.01)* | *1.92 (0.02)* | ***2.26 (0.01)*** |
| **slice**[†] | 53500 | 385 | 0.86 (0.00) | 1.80 (0.00) | 1.86 (0.00) | 0.85 (0.00) | 0.90 (0.00) | 1.80 (0.00) | 1.88 (0.00) | 1.86 (0.00) | **2.02 (0.00)** |
| **keggundirected**[*] | 63608 | 27 | 0.06 (0.01) | 0.06 (0.01) | 0.07 (0.01) | *-75 (4)* | *-4.21 (0.33)* | *-64 (6)* | *2.37 (0.29)* | *-116 (17)* | ***2.98 (0.21)*** |
| 3droad | 434874 | 3 | -0.79 (0.00) | -0.61 (0.01) | -0.58 (0.01) | *-0.71 (0.00)* | *-0.65 (0.00)* | *-0.69 (0.00)* | *-0.55 (0.01)* | *-0.61 (0.00)* | ***-0.48 (0.00)*** |
| song | 515345 | 90 | -1.19 (0.00) | -1.18 (0.00) | -1.18 (0.00) | *-1.15 (0.00)* | *-1.14 (0.00)* | *-1.13 (0.00)* | *-1.10 (0.00)* | *-1.12 (0.00)* | ***-1.07 (0.00)*** |
| buzz | 583250 | 77 | -0.24 (0.00) | -0.15 (0.00) | -0.15 (0.00) | *-0.22 (0.01)* | *-0.23 (0.01)* | *-0.09 (0.01)* | *-0.04 (0.03)* | *-0.09 (0.01)* | ***0.04 (0.00)*** |
| nytaxi | 1420068 | 8 | -1.42 (0.01) | -1.44 (0.01) | -1.42 (0.01) | *-1.09 (0.03)* | ***-0.90 (0.04)*** | *-1.57 (0.04)* | *-1.03 (0.04)* | *-1.74 (0.05)* | *-0.95 (0.07)* |
| **houseelectric**[‡] | 2049280 | 11 | 1.37 (0.00) | 1.41 (0.00) | 1.47 (0.01) | *1.65 (0.03)* | *1.82 (0.01)* | *1.66 (0.05)* | *2.01 (0.00)* | *1.77 (0.06)* | ***2.03 (0.00)*** |
| Median difference from GP baseline | | | 0 | 0.06 | 0.09 | 0.04 | 0.05 | 0.12 | 0.26 | 0.18 | 0.32 |
| Average ranks | | | 2.72 (0.14) | 4.01 (0.14) | 4.84 (0.14) | 4.19 (0.17) | 4.60 (0.16) | 4.84 (0.14) | 6.69 (0.15) | 5.62 (0.16) | 7.49 (0.16) |

modelling the data. Four examples of these datasets are highlighted in **green**[‡]. For these datasets the marginals are non-Gaussian for the latent variable models, and in each case the LV-GP-GP-GP model is the best performing.

**5) On average, the LV-GP-GP-GP model is best performing**. This indicates that the inductive bias of this model is suitable over the broad range of datasets considered. We also compare to Conditional VAE models (Sohn et al., 2015) (see Table 2 in the supplementary material) and find that these models overfit considerably for the smaller datasets and even on the larger datasets do not outperform our deep latent variable models. The marginals of the CVAE models are more Gaussian than our models (see Table 3 in the supplementary material), indicating that these models explain the data through the input mapping and not the complexity of the marginal densities. See Figures 10 and 11 in the supplementary material for the plots of posterior marginals.

## 5. Related Work

If the GP layers are replaced by neural networks our models are equivalent to (conditional) variational autoencoders (VAE) (Kingma and Welling, 2014; Rezende et al., 2014) when used with VI 3.1, or importance weighted autoencoders (IWAEs) (Burda et al., 2016) when used with the IW inference 3.3. Our model can be thought of as a VAE (or IWAE) but with a deep GP for the 'encoder' mapping. Compared to the conditional VAE, the deep GP approach has several advantages: it incorporates model (epistemic) uncertainty, is automatically regularized, and allows a fine-grained control of the properties of the mapping. For example, the lengthscale corresponding to the latent variable

can be tuned to favour complex marginals (a small value) or near Gaussian marginals (a large value). Note that our IW inference is not a simple adaptation of Burda et al. (2016) as we have to perform additional inference over the GPs.

The LV-GP model was proposed by Wang and Neal (2012), and then developed in (Dutordoir et al., 2018) and used for meta learning in Sæmundsson et al. (2018). A version with discrete latent variables was presented in Bodin et al. (2017). The LV-GP without any inputs is known as the Gaussian process latent variable model (Lawrence, 2004), which was used in a semi-supervised setting in Damianou and Lawrence (2015), which is also equivalent to LV-GP.

The deep model of Damianou and Lawrence (2013) is closely related to our model but incorporates the latent variables additively rather than through concatenation (that is, $f(g(x_n) + w_n)$ rather than $f(g([x_n, w_n]))$). In principle, it would be possible to recover our model using the approach of Damianou and Lawrence (2013) in a certain setting of hyperparameters, but in all previous work the kernel hyperparameters were tied within each layer, so this limit was not achievable. Bui et al. (2016) also used this model, with a form of expectation propagation for approximate inference.

The variational inference we have presented (without importance weighting) is not equivalent to that of Damianou and Lawrence (2013). In Damianou and Lawrence (2013) a mean-field variational posterior is used for the noisy corruptions, which may be a poor approximation as there are a priori correlations between these outputs. This mean-field assumption also forces independence between the inputs and outputs of the layers, whereas we make no such assumption.

## 6. Discussion

On a broad range of 1D density estimation tasks we find that our DGP with latent variables outperforms the single-layer and noise-free models, sometimes considerably. Closer investigation reveals that non-Gaussian marginals are readily found by our model, and that the importance-weighted objective improves performance in practice.

Conditional density estimation must balance the complexity of the density with the complexity of the input dependency. The inductive bias of our LV-GP-GP-GP model appears to be suitable for a broad range of datasets we have considered. An advantage of our approach is that the deep models all contain the shallower models as special cases. A layer can be 'turned off' with a single scalar hyperparameter (the kernel variance) set to zero. This is a consequence of the ResNet-inspired (He et al., 2016) use of mean functions. This may explain why we observe empirically that in practice adding depth rarely hurts performance. The latent variables can similarly be 'turned off' if the appropriate lengthscale is large. This may explain why the latent variables models are

rarely outperformed by the noise-free models, even when the marginals are Gaussian.

There are very few hyperparameters in our model to optimize (the kernel parameters and the likelihood variance). This allows us to use the same model across a wide range of data. In the small and medium data regimes we have considered, an unregularized mapping (for example, a neural network as in the conditional VAE model (Sohn et al., 2015)) is likely to overfit, as indeed we have observed (see Table 2 in the supplementary material).

**Limitations**

As depth increases it becomes increasingly difficult to reason about the DGP priors. Our approach is unlikely to recover interpretable features, such as periodicity, and the approaches of Lloyd et al. (2014); Sun et al. (2018) may be more appropriate if an interpretable model is required. The latent variables are also difficult to interpret as their effect is coupled with the GP mappings.

We have only considered 1D latent variables and 1D outputs, and while the extension to higher dimensions is straightforward for training, difficulties may arise in evaluating posterior expectations as our model does not provide a closed-form predictive density and Monte Carlo sampling may have unacceptably high variance. The task of estimating test likelihoods with high-dimensional latent variables is challenging, and techniques described in Wu et al. (2016) may be necessary in higher dimensions.

The inference we have presented is limited by cubic scaling in both $K$ and the number of inducing points. The importance-weighting approach may also suffer from problems of vanishing signal for parameters of $q(w_n)$, as discussed in Rainforth et al. (2018). The doubly reparameterized gradient estimator from Tucker et al. (2019) could be used to alleviate this problem.

## 7. Conclusion

We have presented a novel inference scheme for the deep Gaussian process with latent variables, combining importance weighting with partially collapsed variational inference. We have also developed a variant of the deep Gaussian process model where uncorrelated variables are introduced as latent covariates rather than process noise. We have shown empirically that latent variables models deep models outperform the noise-free deep GP on a range of data, and also that our importance-weighted inference delivers an advantage over variational inference in practice.

## Acknowledgements

## References

Mauricio A Alvarez, Lorenzo Rosasco, Neil D Lawrence, et al. Kernels for Vector-valued Functions: A Review. *Foundations and Trends in Machine Learning*, 2012.

Erik Bodin, Neill D Campbell, and Carl H Ek. Latent Gaussian Process Regression. *arXiv:1707.05534*, 2017.

Thang Bui, Daniel Hernández-Lobato, Jose Hernandez-Lobato, Yingzhen Li, and Richard Turner. Deep Gaussian Processes for Regression using Approximate Expectation Propagation. *International Conference on Machine Learning*, 2016.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance Weighted Autoencoders. *International Conference on Learning Representations*, 2016.

Ching-An Cheng and Byron Boots. Incremental Variational Sparse Gaussian Process Regression. *Advances in Neural Information Processing Systems*, 2016.

Zhenwen Dai, Andreas Damianou, Javier González, and Neil Lawrence. Variational Auto-encoded Deep Gaussian Processes. *International Conference on Learning Representations*, 2015.

Zhenwen Dai, Mauricio A Álvarez, and Neil Lawrence. Efficient Modeling of Latent Information in Supervised Learning using Gaussian Processes. *Advances in Neural Information Processing Systems*, 2017.

Andreas Damianou and Neil Lawrence. Deep gaussian processes. *Artificial Intelligence and Statistics*, 2013.

Andreas Damianou and Neil D Lawrence. Semi-described and Semi-supervised Learning with Gaussian Processes. *arXiv:1509.01168*, 2015.

Justin Domke and Daniel Sheldon. Importance Weighting and Varational Inference. *Advances in Neural Information Processing Systems*, 2018.

Matthew M Dunlop, Mark Girolami, Andrew M Stuart, and Aretha L Teckentrup. How Deep Are Deep Gaussian Processes? *Journal of Machine Learning Research*, 2018.

Vincent Dutordoir, Hugh Salimbeni, Marc P Deisenroth, and James Hensman. Gaussian Process Conditional Density Estimation. *Advances in Neural Information Processing Systems*, 2018.

David Duvenaud, Oren Rippel, Ryan Adams, and Zoubin Ghahramani. Avoiding Pathologies in Very Deep Networks. *Artificial Intelligence and Statistics*, 2014.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. *Artificial Intelligence and Statistics*, 2010.

Marton Havasi, José Miguel Hernández Lobato, and Juan José Murillo Fuentes. Inference in Deep Gaussian Processes using Stochastic Gradient Hamiltonian Monte Carlo. *Advances in Neural Information Processing Systems*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *The IEEE conference on computer vision and pattern recognition*, 2016.

Ali Hebbal, Loic Brevault, Mathieu Balesdent, El-Ghazali Talbi, and Nouredine Melab. Bayesian Optimization using Deep Gaussian Processes. *arXiv:1905.03350*, 2019.

James Hensman, Magnus Rattray, and Neil D Lawrence. Fast Variational Inference in the Conjugate Exponential Family. *Advances in Neural Information Processing Systems*, 2012.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian Processes for Big Data. *Uncertainty in Artificial Intelligence*, 2013.

Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*, 2015.

Diederik P Kingma and Max Welling. Auto-encoding Variational Bayes. *International Conference for Learning Representations*, 2014.

Neil D Lawrence. Gaussian Process Latent Variable Models for Visualisation of High Dimensional Data. *Advances in Neural Information Processing Systems*, 2004.

James Robert Lloyd, David K Duvenaud, Roger B Grosse, Joshua B Tenenbaum, and Zoubin Ghahramani. Automatic construction and natural-language description of nonparametric regression models. *The AAAI Conference on Artificial Intelligence*, 2014.

Alexander Matthews, James Hensman, Turner Richard, and Zoubin Ghahramani. On Sparse Variational Methods and the Kullback-Leibler Divergence between Stochastic Processes. *Artificial Intelligence and Statistics*, 2016.

Alexander G de G Matthews, Mark van der Wilk, Tom Nickson, Keisuke Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman.

GPflow: a Gaussian Process Library using TensorFlow. *The Journal of Machine Learning Research*, 2017.

Radford M Neil. Regression and Classification using Gaussian Process Priors. *Bayesian Statistics*, 1998.

Tom Rainforth, Adam R Kosiorek, Tuan Anh Le, Chris J Maddison, Maximilian Igl, Frank Wood, and Yee Whye Teh. Tighter variational bounds are not necessarily better. *International Conference on Machine Learning*, 2018.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *International Conference on Machine Learning*, 2014.

Steindór Sæmundsson, Katja Hofmann, and Marc Peter Deisenroth. Meta Reinforcement Learning with Latent Variable Gaussian Processes. *Uncertainty in Artificial Intelligence*, 2018.

Hugh Salimbeni and Marc P Deisenroth. Doubly Stochastic Variational Inference for Deep Gaussian Processes. *Advances in Neural Information Processing Systems*, 2017.

Hugh Salimbeni, Stefanos Eleftheriadis, and James Hensman. Natural Gradients in Practice: Non-Conjugate Variational Inference in Gaussian Process Models. *Artificial Intelligence and Statistics*, 2018.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in Neural Information Processing Systems*, 2015.

Shengyang Sun, Guodong Zhang, Chaoqi Wang, Wenyuan Zeng, Jiaman Li, and Roger Grosse. Differentiable compositional kernel learning for gaussian processes. *International Conference on Machine Learning*, 2018.

Michalis Titsias. Variational Learning of Inducing Variables in Sparse Gaussian Processes. *Artificial Intelligence and Statistics*, 2009.

Michalis Titsias and Neil D Lawrence. Bayesian Gaussian Process Latent Variable Model. *Artificial Intelligence and Statistics*, 2010.

George Tucker, Dieterich Lawson, Shixiang Gu, and Chris J Maddison. Doubly Reparameterized Gradient Estimators for Monte Carlo Objectives. *International Conference on Learning Representations*, 2019.

Chunyi Wang and Radford Neal. Gaussian Process Regression with Heteroscedastic or Non-Gaussian Residuals. *arXiv:1212.6246*, 2012.

Yuhuai Wu, Yuri Burda, Ruslan Salakhutdinov, and Roger Grosse. On the Quantitative Analysis of Decoder-based Generative Models. *International Conference on Learning Representations*, 2016.