THE VARIATIONAL GAUSSIAN PROCESS

Dustin Tran
Harvard University
dtran@g.harvard.edu

Rajesh Ranganath
Princeton University
rajeshr@cs.princeton.edu

David M. Blei
Columbia University
david.blei@columbia.edu

ABSTRACT

Variational inference is a powerful tool for approximate inference, and it has been recently applied for representation learning with deep generative models. We develop the *variational Gaussian process (*VGP*)*, a Bayesian nonparametric variational family, which adapts its shape to match complex posterior distributions. The VGP generates approximate posterior samples by generating latent inputs and warping them through random non-linear mappings; the distribution over random mappings is learned during inference, enabling the transformed outputs to adapt to varying complexity. We prove a universal approximation theorem for the VGP, demonstrating its representative power for learning any model. For inference we present a variational objective inspired by auto-encoders and perform black box inference over a wide class of models. The VGP achieves new state-of-the-art results for unsupervised learning, inferring models such as the deep latent Gaussian model and the recently proposed DRAW.

1 INTRODUCTION

Variational inference is a powerful tool for approximate posterior inference. The idea is to posit a family of distributions over the latent variables and then find the member of that family closest to the posterior. Originally developed in the 1990s (Hinton & Van Camp, 1993; Waterhouse et al., 1996; Jordan et al., 1999), variational inference has enjoyed renewed interest around developing scalable optimization for large datasets (Hoffman et al., 2013), deriving generic strategies for easily fitting many models (Ranganath et al., 2014), and applying neural networks as a flexible parametric family of approximations (Kingma & Welling, 2014; Rezende et al., 2014). This research has been particularly successful for computing with deep Bayesian models (Neal, 1990; Ranganath et al., 2015a), which require inference of a complex posterior distribution (Hinton et al., 2006).

Classical variational inference typically uses the mean-field family, where each latent variable is independent and governed by its own variational distribution. While convenient, the strong independence limits learning deep representations of data. Newer research aims toward richer families that allow dependencies among the latent variables. One way to introduce dependence is to consider the variational family itself as a model of the latent variables (Lawrence, 2000; Ranganath et al., 2015b). These *variational models* naturally extend to Bayesian hierarchies, which retain the mean-field "likelihood" but introduce dependence through variational latent variables.

In this paper we develop a powerful new variational model—the variational Gaussian process (VGP). The VGP is a Bayesian nonparametric variational model; its complexity grows efficiently and towards *any* distribution, adapting to the inference problem at hand. We highlight three main contributions of this work:

1. We prove a universal approximation theorem: under certain conditions, the VGP can capture any continuous posterior distribution—it is a variational family that can be specified to be as expressive as needed.
2. We derive an efficient stochastic optimization algorithm for variational inference with the VGP. Our algorithm can be used in a wide class of models. Inference with the VGP is a black box variational method (Ranganath et al., 2014).
3. We study the VGP on standard benchmarks for unsupervised learning, applying it to perform inference in deep latent Gaussian models (Rezende et al., 2014) and DRAW (Gregor et al., 2015), a latent attention model. For both models, we report the best results to date.

Technical summary. Generative models hypothesize a distribution of observations $\mathbf{x}$ and latent variables $\mathbf{z}$, $p(\mathbf{x}, \mathbf{z})$. Variational inference posits a family of the latent variables $q(\mathbf{z}; \boldsymbol{\lambda})$ and tries to find the variational parameters $\boldsymbol{\lambda}$ that are closest in KL divergence to the posterior. When we use a variational model, $q(\mathbf{z}; \boldsymbol{\lambda})$ itself might contain variational latent variables; these are implicitly marginalized out in the variational family (Ranganath et al., 2015b).

The VGP is a flexible variational model. It draw inputs from a simple distribution, warps those inputs through a non-linear mapping, and then uses the output of the mapping to govern the distribution of the latent variables $\mathbf{z}$. The non-linear mapping is itself a random variable, constructed from a Gaussian process. The VGP is inspired by ideas from both the Gaussian process latent variable model (Lawrence, 2005) and Gaussian process regression (Rasmussen & Williams, 2006).

The variational parameters of the VGP are the kernel parameters for the Gaussian process and a set of *variational data*, which are input-output pairs. The variational data is crucial: it anchors the non-linear mappings at given inputs and outputs. It is through these parameters that the VGP learns complex representations. Finally, given data $\mathbf{x}$, we use stochastic optimization to find the variational parameters that minimize the KL divergence to the model posterior.

2 Variational Gaussian Process

Variational models introduce latent variables to the variational family, providing a rich construction for posterior approximation (Ranganath et al., 2015b). Here we introduce the variational Gaussian process (VGP), a Bayesian nonparametric variational model that is based on the Gaussian process. The Gaussian process (GP) provides a class of latent variables that lets us capture downstream distributions with varying complexity.

We first review variational models and Gaussian processes. We then outline the mechanics of the VGP and prove that it is a universal approximator.

2.1 Variational models

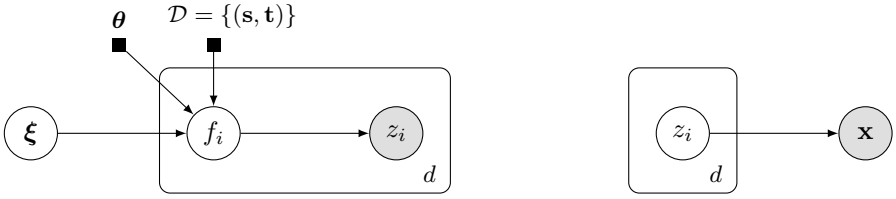
Let $p(\mathbf{z} \mid \mathbf{x})$ denote a posterior distribution over d latent variables $\mathbf{z} = (z_1, \ldots, z_d)$ conditioned on a data set $\mathbf{x}$. For a family of distributions $q(\mathbf{z}; \boldsymbol{\lambda})$ parameterized by $\boldsymbol{\lambda}$, variational inference seeks to minimize the divergence $\mathrm{KL}(q(\mathbf{z}; \boldsymbol{\lambda}) \,\|\, p(\mathbf{z} \mid \mathbf{x}))$. This is equivalent to maximizing the evidence lower bound (ELBO) (Wainwright & Jordan, 2008). The ELBO can be written as a sum of the expected log likelihood of the data and the KL divergence between the variational distribution and the prior,

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})}[\log p(\mathbf{x} \mid \mathbf{z})] - \mathrm{KL}(q(\mathbf{z}; \boldsymbol{\lambda}) \| p(\mathbf{z})). \tag{1}$$

Traditionally, variational inference considers a tractable family of distributions with analytic forms for its density. A common specification is a fully factorized distribution $\prod_i q(z_i; \lambda_i)$, also known as the mean-field family. While mean-field families lead to efficient computation, they limit the expressiveness of the approximation.

The variational family of distributions can be interpreted as a model of the latent variables $\mathbf{z}$, and it can be made richer by introducing new latent variables. Hierarchical variational models consider distributions specified by a variational prior of the mean-field parameters $q(\boldsymbol{\lambda}; \boldsymbol{\theta})$ and a factorized "likelihood" $\prod_i q(z_i \mid \lambda_i)$. This specifies the variational model,

$$q(\mathbf{z}; \boldsymbol{\theta}) = \int \Big[\prod_i q(z_i \mid \lambda_i)\Big] q(\boldsymbol{\lambda}; \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\lambda}, \tag{2}$$



(a) VARIATIONAL MODEL **(b)** GENERATIVE MODEL

Figure 1: (a) Graphical model of the variational Gaussian process. The VGP generates samples of latent variables $\mathbf{z}$ by evaluating random non-linear mappings of latent inputs $\boldsymbol{\xi}$, and then drawing mean-field samples parameterized by the mapping. These latent variables aim to follow the posterior distribution for a generative model **(b)**, conditioned on data $\mathbf{x}$.

which is governed by prior hyperparameters $\boldsymbol{\theta}$. Hierarchical variational models are richer than classical variational families—their expressiveness is determined by the complexity of the prior $q(\boldsymbol{\lambda})$. Many expressive variational approximations can be viewed under this construct (Saul & Jordan, 1996; Jaakkola & Jordan, 1998; Rezende & Mohamed, 2015; Tran et al., 2015).

2.2 GAUSSIAN PROCESSES

We now review the Gaussian process (GP) (Rasmussen & Williams, 2006). Consider a data set of m source-target pairs $\mathcal{D} = \{(\mathbf{s}_n, \mathbf{t}_n)\}_{n=1}^m$, where each source $\mathbf{s}_n$ has c covariates paired with a multi-dimensional target $\mathbf{t}_n \in \mathbb{R}^d$. We aim to learn a function over all source-target pairs, $\mathbf{t}_n = f(\mathbf{s}_n)$, where $f : \mathbb{R}^c \to \mathbb{R}^d$ is unknown. Let the function f decouple as $f = (f_1, \ldots, f_d)$, where each $f_i : \mathbb{R}^c \to \mathbb{R}$. GP regression estimates the functional form of f by placing a prior,

$$p(f) = \prod_{i=1}^{d} \mathcal{GP}(f_i; \mathbf{0}, \mathbf{K}_{ss}),$$

where $\mathbf{K}_{ss}$ denotes a covariance function $k(\mathbf{s}, \mathbf{s}')$ evaluated over pairs of inputs $\mathbf{s}, \mathbf{s}' \in \mathbb{R}^c$. In this paper, we consider automatic relevance determination (ARD) kernels

$$k(\mathbf{s}, \mathbf{s}') = \sigma_{\text{ARD}}^2 \exp\Big(-\frac{1}{2} \sum_{j=1}^{c} \omega_j (s_j - s_j')^2 \Big), \tag{3}$$

with parameters $\boldsymbol{\theta} = (\sigma_{\text{ARD}}^2, \omega_1, \ldots, \omega_c)$. The weights ω_j tune the importance of each dimension. They can be driven to zero during inference, leading to automatic dimensionality reduction.

Given data $\mathcal{D}$, the conditional distribution of the GP forms a distribution over mappings which interpolate between input-output pairs,

$$p(f \mid \mathcal{D}) = \prod_{i=1}^{d} \mathcal{GP}(f_i; \mathbf{K}_{\boldsymbol{\xi}s}\mathbf{K}_{ss}^{-1}\mathbf{t}_i, \mathbf{K}_{\boldsymbol{\xi}\boldsymbol{\xi}} - \mathbf{K}_{\boldsymbol{\xi}s}\mathbf{K}_{ss}^{-1}\mathbf{K}_{\boldsymbol{\xi}s}^\top). \tag{4}$$

Here, $\mathbf{K}_{\boldsymbol{\xi}s}$ denotes the covariance function $k(\boldsymbol{\xi}, \mathbf{s})$ for an input $\boldsymbol{\xi}$ and over all data inputs $\mathbf{s}_n$, and $\mathbf{t}_i$ represents the i^{th} output dimension.

2.3 VARIATIONAL GAUSSIAN PROCESSES

We describe the variational Gaussian process (VGP), a Bayesian nonparametric variational model that admits arbitrary structures to match posterior distributions. The VGP generates $\mathbf{z}$ by generating latent inputs, warping them with random non-linear mappings, and using the warped inputs as parameters to a mean-field distribution. The random mappings are drawn conditional on "variational data," which are variational parameters. We will show that the VGP enables samples from the mean-field to follow arbitrarily complex posteriors.

The VGP specifies the following generative process for posterior latent variables $\mathbf{z}$:

1. Draw latent input $\boldsymbol{\xi} \in \mathbb{R}^c$: $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
2. Draw non-linear mapping $f : \mathbb{R}^c \to \mathbb{R}^d$ conditioned on $\mathcal{D}$: $f \sim \prod_{i=1}^d \mathcal{GP}(\mathbf{0}, \mathbf{K}_{\boldsymbol{\xi}\boldsymbol{\xi}}) \,|\, \mathcal{D}$.
3. Draw approximate posterior samples $\mathbf{z} \in \text{supp}(p)$: $\mathbf{z} = (z_1, \ldots, z_d) \sim \prod_{i=1}^d q(f_i(\boldsymbol{\xi}))$.

Figure 1 displays a graphical model for the VGP. Here, $\mathcal{D} = \{(\mathbf{s}_n, \mathbf{t}_n)\}_{n=1}^m$ represents variational data, comprising input-output pairs that are parameters to the variational distribution. Marginalizing over all latent inputs and non-linear mappings, the VGP is

$$q_{\text{VGP}}(\mathbf{z}; \boldsymbol{\theta}, \mathcal{D}) = \iint \left[\prod_{i=1}^d q(z_i \,|\, f_i(\boldsymbol{\xi}))\right] \left[\prod_{i=1}^d \mathcal{GP}(f_i; \mathbf{0}, \mathbf{K}_{\boldsymbol{\xi}\boldsymbol{\xi}}) \,|\, \mathcal{D}\right] \mathcal{N}(\boldsymbol{\xi}; \mathbf{0}, \mathbf{I}) \,\mathrm{d}f \,\mathrm{d}\boldsymbol{\xi}. \tag{5}$$

The VGP is parameterized by kernel hyperparameters $\boldsymbol{\theta}$ and variational data.

As a variational model, the VGP forms an infinite ensemble of mean-field distributions. A mean-field distribution is given in the first term of the integrand above. It is *conditional* on a fixed function $f(\cdot)$ and input $\boldsymbol{\xi}$; the d outputs $f_i(\boldsymbol{\xi}) = \lambda_i$ are the mean-field's parameters. The VGP is a form of a hierarchical variational model (Eq.2) (Ranganath et al., 2015b). It places a continuous Bayesian nonparametric prior over mean-field parameters.

Unlike the mean-field, the VGP can capture correlation between the latent variables. The reason is that it evaluates the d independent GP draws at the same latent input $\boldsymbol{\xi}$. This induces correlation between their outputs, the mean-field parameters, and thus also correlation between the latent variables. Further, the VGP is flexible. The complex non-linear mappings drawn from the GP allow it to capture complex discrete and continuous posteriors.

We emphasize that the VGP needs variational data. Unlike typical GP regression, there are no observed data available to learn a distribution over non-linear mappings of the latent variables $\mathbf{z}$. Thus the "data" are variational parameters that appear in the conditional distribution of f in Eq.4. They anchor the random non-linear mappings at certain input-ouput pairs. When optimizing the VGP, the learned variational data enables finds a distribution of the latent variables that closely follows the posterior.

2.4 Universal approximation theorem

To understand the capacity of the VGP for representing complex posterior distributions, we analyze the role of the Gaussian process. For simplicity, suppose the latent variables $\mathbf{z}$ are real-valued, and the VGP treats the output of the function draws from the GP as posterior samples. Consider the optimal function f^*, which is the transformation such that when we draw $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and calculate $\mathbf{z} = f^*(\boldsymbol{\xi})$, the resulting distribution of $\mathbf{z}$ *is* the posterior distribution.

An explicit construction of f^* exists if the dimension of the latent input $\boldsymbol{\xi}$ is equal to the number of latent variables. Let P^{-1} denote the inverse posterior CDF and Φ the standard normal CDF. Using techniques common in copula literature (Nelsen, 2006), the optimal function is

$$f^*(\boldsymbol{\xi}) = P^{-1}(\Phi(\xi_1), \ldots, \Phi(\xi_d)).$$

Imagine generating samples $\mathbf{z}$ using this function. For latent input $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the standard normal CDF Φ applies the probability integral transform: it squashes ξ_i such that its output $u_i = \Phi(\xi_i)$ is uniformly distributed on $[0, 1]$. The inverse posterior CDF then transforms the uniform random variables $P^{-1}(u_1, \ldots, u_d) = \mathbf{z}$ to follow the posterior. The function produces exact posterior samples.

In the VGP, the random function interpolates the values in the variational data, which are optimized to minimize the KL divergence. Thus, during inference, the distribution of the GP learns to concentrate around this optimal function. This perspective provides intuition behind the following result.

Theorem 1 (Universal approximation)**.** *Let $q(\mathbf{z}; \boldsymbol{\theta}, \mathcal{D})$ denote the variational Gaussian process. Consider a posterior distribution $p(\mathbf{z} \,|\, \mathbf{x})$ with a finite number of latent variables and continuous quantile function (inverse CDF). There exists a sequence of parameters $(\boldsymbol{\theta}_k, \mathcal{D}_k)$ such that*

$$\lim_{k \to \infty} \text{KL}(q(\mathbf{z}; \boldsymbol{\theta}_k, \mathcal{D}_k) \,\|\, p(\mathbf{z} \,|\, \mathbf{x})) = 0.$$

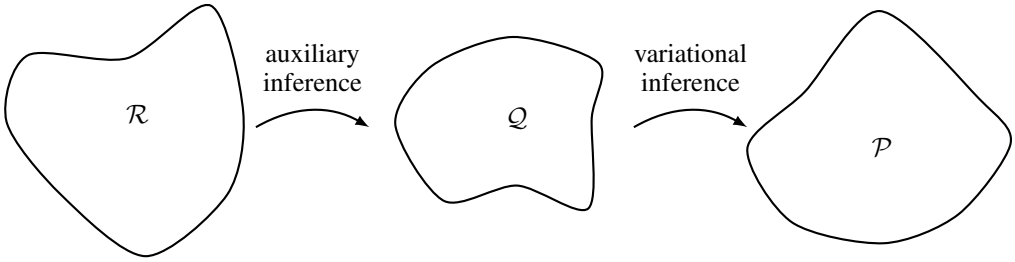


Figure 2: Sequence of domain mappings during inference, from variational latent variable space $\mathcal{R}$ to posterior latent variable space $\mathcal{Q}$ to data space $\mathcal{P}$. We perform variational inference in the posterior space and auxiliary inference in the variational space.

See Appendix B for a proof. Theorem 1 states that any posterior distribution with strictly positive density can be represented by a VGP. Thus the VGP is a flexible model for learning posterior distributions.

3 Black box inference

We derive an algorithm for black box inference over a wide class of generative models.

3.1 Variational objective

The original ELBO (Eq.1) is analytically intractable due to the log density, $\log q_{\text{VGP}}(\mathbf{z})$ (Eq.5). To address this, we present a tractable variational objective inspired by auto-encoders (Kingma & Welling, 2014).

A tractable lower bound to the model evidence $\log p(\mathbf{x})$ can be derived by subtracting an expected KL divergence term from the ELBO,

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q_{\text{VGP}}}[\log p(\mathbf{x} \mid \mathbf{z})] - \text{KL}(q_{\text{VGP}}(\mathbf{z}) \| p(\mathbf{z})) - \mathbb{E}_{q_{\text{VGP}}}\Big[\text{KL}(q(\boldsymbol{\xi}, f \mid \mathbf{z}) \| r(\boldsymbol{\xi}, f \mid \mathbf{z}))\Big],$$

where $r(\boldsymbol{\xi}, f \mid \mathbf{z})$ is an auxiliary model (we describe r in the next subsection). Various versions of this objective have been considered in the literature (Jaakkola & Jordan, 1998; Agakov & Barber, 2004), and it has been recently revisited by Salimans et al. (2015) and Ranganath et al. (2015b). We perform variational inference in the posterior latent variable space, minimizing $\text{KL}(q \| p)$ to learn the variational model; for this to occur we perform auxiliary inference in the variational latent variable space, minimizing $\text{KL}(q \| r)$ to learn an auxiliary model. See Figure 2.

Unlike previous approaches, we rewrite this variational objective to connect to auto-encoders:

$$\begin{aligned}\widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{q_{\text{VGP}}}[\log p(\mathbf{x} \mid \mathbf{z})] - \mathbb{E}_{q_{\text{VGP}}}\Big[\text{KL}(q(\mathbf{z} \mid f(\boldsymbol{\xi})) \| p(\mathbf{z}))\Big] \\ - \mathbb{E}_{q_{\text{VGP}}}\Big[\text{KL}(q(f \mid \boldsymbol{\xi}; \boldsymbol{\theta}) \| r(f \mid \boldsymbol{\xi}, \mathbf{z}; \boldsymbol{\phi})) + \log q(\boldsymbol{\xi}) - \log r(\boldsymbol{\xi} \mid \mathbf{z})\Big],\end{aligned} \tag{6}$$

where the KL divergences are now taken over tractable distributions (see Appendix C). In auto-encoder parlance, we maximize the expected negative reconstruction error, regularized by two terms: an expected divergence between the variational model and the original model's prior, and an expected divergence between the auxiliary model and the variational model's prior. This is simply a nested instantiation of the variational auto-encoder bound (Kingma & Welling, 2014): a divergence between the inference model and a prior is taken as regularizers on both the posterior and variational spaces. This interpretation justifies the previously proposed bound for variational models; as we shall see, it also enables lower variance gradients during stochastic optimization.

3.2 Auto-encoding variational models

An inference network provide a flexible parameterization of approximating distributions as used in Helmholtz machines (Hinton & Zemel, 1994), deep Boltzmann machines (Salakhutdinov &

Larochelle, 2010), and variational auto-encoders (Kingma & Welling, 2014; Rezende et al., 2014). It replaces local variational parameters with global parameters coming from a neural network. For latent variables $\mathbf{z}_n$ (which correspond to a data point $\mathbf{x}_n$), an inference network specifies a neural network which takes $\mathbf{x}_n$ as input and its local variational parameters $\boldsymbol{\lambda}_n$ as output. This amortizes inference by only defining a set of global parameters.

To auto-encode the VGP we specify inference networks to parameterize both the variational and auxiliary models:

$$\mathbf{x}_n \mapsto q(\mathbf{z}_n \mid \mathbf{x}_n; \boldsymbol{\theta}_n), \qquad \mathbf{x}_n, \mathbf{z}_n \mapsto r(\boldsymbol{\xi}_n, f_n \mid \mathbf{x}_n, \mathbf{z}_n; \boldsymbol{\phi}_n).$$

Formally, the output of these mappings are the parameters $\boldsymbol{\theta}_n$ and $\boldsymbol{\phi}_n$ respectively. We write the output as distributions above to emphasize that these mappings are a (global) parameterization of the variational model q and auxiliary model r. The local variational parameters $\boldsymbol{\theta}_n$ for q are the variational data $\mathcal{D}_n$. The auxiliary model r is specified as a fully factorized Gaussian with local variational parameters $\boldsymbol{\phi}_n = (\boldsymbol{\mu}_n \in \mathbb{R}^{c+d}, \boldsymbol{\sigma}_n^2 \in \mathbb{R}^{c+d})$. [1]

3.3 Stochastic optimization

We maximize the variational objective $\widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi})$ over both $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, where $\boldsymbol{\theta}$ newly denotes both the kernel hyperparameters and the inference network's parameters for the VGP, and $\boldsymbol{\phi}$ denotes the inference network's parameters for the auxiliary model. Following the black box methods, we write the gradient as an expectation and apply stochastic approximations (Robbins & Monro, 1951), sampling from the variational model and evaluating noisy gradients.

First, we reduce variance of the stochastic gradients by analytically deriving any tractable expectations. The KL divergence between $q(\mathbf{z} \mid f(\boldsymbol{\xi}))$ and $p(\mathbf{z})$ is commonly used to reduce variance in traditional variational auto-encoders: it is analytic for deep generative models such as the deep latent Gaussian model (Rezende et al., 2014) and deep recurrent attentive writer (Gregor et al., 2015). The KL divergence between $r(f \mid \boldsymbol{\xi}, \mathbf{z})$ and $q(f \mid \boldsymbol{\xi})$ is analytic as the distributions are both Gaussian. The difference $\log q(\boldsymbol{\xi}) - \log r(\boldsymbol{\xi} \mid \mathbf{z})$ is simply a difference of Gaussian log densities. See Appendix C for more details.

To derive black box gradients, we can first reparameterize the VGP, separating noise generation of samples from the parameters in its generative process (Kingma & Welling, 2014; Rezende et al., 2014). The GP easily enables reparameterization: for latent inputs $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the transformation $\mathbf{f}(\boldsymbol{\xi}; \boldsymbol{\theta}) = \mathbf{L}\boldsymbol{\xi} + \mathbf{K}_{\boldsymbol{\xi}s}\mathbf{K}_{ss}^{-1}\mathbf{t}_i$ is a location-scale transform, where $\mathbf{L}\mathbf{L}^\top = \mathbf{K}_{\boldsymbol{\xi}\boldsymbol{\xi}} - \mathbf{K}_{\boldsymbol{\xi}s}\mathbf{K}_{ss}^{-1}\mathbf{K}_{\boldsymbol{\xi}s}^\top$. This is equivalent to evaluating $\boldsymbol{\xi}$ with a random mapping from the GP. Suppose the mean-field $q(\mathbf{z} \mid f(\boldsymbol{\xi}))$ is also reparameterizable, and let $\boldsymbol{\epsilon} \sim w$ such that $\mathbf{z}(\boldsymbol{\epsilon}; \mathbf{f})$ is a function of $\boldsymbol{\xi}$ whose output $\mathbf{z} \sim q(\mathbf{z} \mid f(\boldsymbol{\xi}))$. This two-level reparameterization is equivalent to the generative process for $\mathbf{z}$ outlined in Section 2.3.

We now rewrite the variational objective as

$$\begin{aligned}\widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{\mathcal{N}(\boldsymbol{\xi})}\Big[\mathbb{E}_{w(\boldsymbol{\epsilon})}\Big[\log p(\mathbf{x} \mid \mathbf{z}(\boldsymbol{\epsilon}; \mathbf{f}))\Big] - \mathrm{KL}(q(\mathbf{z} \mid \mathbf{f}) \| p(\mathbf{z}))\Big] \\ - \mathbb{E}_{\mathcal{N}(\boldsymbol{\xi})}\Big[\mathbb{E}_{w(\boldsymbol{\epsilon})}\Big[\mathrm{KL}(q(f \mid \boldsymbol{\xi}; \boldsymbol{\theta}) \| r(f \mid \boldsymbol{\xi}, \mathbf{z}(\boldsymbol{\epsilon}; \mathbf{f}); \boldsymbol{\phi})) + \log q(\boldsymbol{\xi}) - \log r(\boldsymbol{\xi} \mid \mathbf{z}(\boldsymbol{\epsilon}; \mathbf{f}))\Big]\Big].\end{aligned} \tag{7}$$

Eq.7 enables gradients to move inside the expectations and backpropagate over the nested reparameterization. Thus we can take unbiased stochastic gradients, which exhibit low variance due to both the analytic KL terms and reparameterization. The gradients are derived in Appendix D, including the case when the first KL is analytically intractable.

We outline the method in Algorithm 1. For massive data, we apply subsampling on $\mathbf{x}$ (Hoffman et al., 2013). For gradients of the model log-likelihood, we employ convenient differentiation tools such as those in Stan and Theano (Carpenter et al., 2015; Bergstra et al., 2010). For non-differentiable latent variables $\mathbf{z}$, or mean-field distributions without efficient reparameterizations, we apply the black box gradient estimator from Ranganath et al. (2014) to take gradients of the inner expectation.

[1] We let the kernel hyperparameters of the VGP be fixed across data points. Note also that unique from other auto-encoder approaches, we let r's inference network take both $\mathbf{x}_n$ and $\mathbf{z}_n$ as input: this avoids an explicit specification of the conditional distribution $r(\boldsymbol{\epsilon}, f \mid \mathbf{z})$, which may be difficult to model. This idea was first suggested (but not implemented) in Ranganath et al. (2015b).

Algorithm 1: Black box inference with a variational Gaussian process

Input: Model $p(\mathbf{x}, \mathbf{z})$, Mean-field family $\prod_i q(\mathbf{z}_i \mid f_i(\boldsymbol{\xi}))$.

Output: Variational and auxiliary parameters $(\boldsymbol{\theta}, \boldsymbol{\phi})$.

Initialize $(\boldsymbol{\theta}, \boldsymbol{\phi})$ randomly.

while not converged **do**

Draw noise samples $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\boldsymbol{\epsilon} \sim w$.

Parameterize variational samples $\mathbf{z} = \mathbf{z}(\boldsymbol{\epsilon}; f(\boldsymbol{\xi}))$, $f(\boldsymbol{\xi}) = \mathbf{f}(\boldsymbol{\xi}; \boldsymbol{\theta})$.

Update $(\boldsymbol{\theta}, \boldsymbol{\phi})$ with stochastic gradients $\nabla_{\boldsymbol{\theta}} \widetilde{\mathcal{L}}$, $\nabla_{\boldsymbol{\phi}} \widetilde{\mathcal{L}}$.

end

3.4 Computational and storage complexity

The algorithm has $\mathcal{O}(d + m^3 + LH^2)$ complexity, where d is the number of latent variables, m is the size of the variational data, and L is the number of layers of the neural networks with H the average hidden layer size. In particular, the algorithm is linear in the number of latent variables, which is competitive with other variational inference methods. The number of variational and auxiliary parameters has $\mathcal{O}(c + LH)$ complexity; this complexity comes from storing the kernel hyperparameters and the neural network parameters.

Unlike most GP literature, we require no low rank constraints, such as the use of inducing variables for scalable computation (Quiñonero-Candela & Rasmussen, 2005). The variational data serve a similar purpose, but inducing variables reduce the rank of a (fixed) kernel matrix; the variational data directly determine the kernel matrix and thus the kernel matrix is not fixed. Although we haven't found it necessary in practice, see Appendix E for scaling the size of variational data.

4 Related work

Recently, there has been interest in applying parametric transformations for approximate inference. Parametric transformations of random variables induce a density in the transformed space, with a Jacobian determinant that accounts for how the transformation warps unit volumes. Kucukelbir et al. (2016) consider this viewpoint for automating inference, in which they posit a transformation from the standard normal to a possibly constrained latent variable space. In general, however, calculating the Jacobian determinant incurs a costly $\mathcal{O}(d^3)$ complexity, cubic in the number of latent variables. Dinh et al. (2015) consider volume-preserving transformations which avoid calculating Jacobian determinants. Salimans et al. (2015) consider volume-preserving transformations defined by Markov transition operators. Rezende & Mohamed (2015) consider a slightly broader class of parametric transformations, with Jacobian determinants having at most $\mathcal{O}(d)$ complexity.

Instead of specifying a parametric class of mappings, the VGP posits a Bayesian nonparametric prior over all continuous mappings. The VGP can recover a certain class of parametric transformations by using kernels which induce a prior over that class. In the context of the VGP, the GP is an infinitely wide feedforward network which warps latent inputs to mean-field parameters. Thus, the VGP offers complete flexibility on the space of mappings—there are no restrictions such as invertibility or linear complexity—and is fully Bayesian. Further, it is a hierarchical variational model, using the GP as a variational prior over mean-field parameters (Ranganath et al., 2015b). This enables inference over both discrete and continuous latent variable models.

In addition to its flexibility over parametric methods, the VGP is more computationally efficient. Parametric methods must consider transformations with Jacobian determinants of at most $\mathcal{O}(d)$ complexity. This restricts the flexibility of the mapping and therefore the flexibility of the variational model (Rezende & Mohamed, 2015). In comparison, the distribution of outputs using a GP prior does not require any Jacobian determinants (following Eq.4); instead it requires auxiliary inference for learning variational latent variables (which is fast). Further, unlike discrete Bayesian

Model	$-\log p(\mathbf{x})$	$\leq$
DLGM + VAE [1]		86.76
DLGM + HVI (8 leapfrog steps) [2]	85.51	88.30
DLGM + NF ($k = 80$) [3]		85.10
EoNADE-5 2hl (128 orderings) [4]	84.68	
DBN 2hl [5]	84.55	
DARN 1hl [6]	84.13	
Convolutional VAE + HVI [2]	81.94	83.49
DLGM 2hl + IWAE ($k = 50$) [1]		82.90
DRAW [7]		80.97
DLGM 1hl + VGP		83.64
DLGM 2hl + VGP		81.90
DRAW + VGP		**80.11**

Table 1: Negative predictive log-likelihood for binarized MNIST. Previous best results are [1] (Burda et al., 2016), [2] (Salimans et al., 2015), [3] (Rezende & Mohamed, 2015), [4] (Raiko et al., 2014), [5] (Murray & Salakhutdinov, 2009), [6] (Gregor et al., 2014), [7] (Gregor et al., 2015).

nonparametric priors such as an infinite mixture of mean-field distributions, the GP enables black box inference with lower variance gradients—it applies a location-scale transform for reparameterization and has analytically tractable KL terms.

Transformations, which convert samples from a tractable distribution to the posterior, is a classic technique in Bayesian inference. It was first studied in Monte Carlo methods, where it is core to the development of methods such as path sampling, annealed importance sampling, and sequential Monte Carlo (Gelman & Meng, 1998; Neal, 1998; Chopin, 2002). These methods can be recast as specifying a discretized mapping f_t for times $t_0 < \ldots < t_k$, such that for draws $\boldsymbol{\xi}$ from the tractable distribution, $f_{t_0}(\boldsymbol{\xi})$ outputs the same samples and $f_{t_k}(\boldsymbol{\xi})$ outputs exact samples following the posterior. By applying the sequence in various forms, the transformation bridges the tractable distribution to the posterior. Specifying a good transformation—termed "schedule" in the literature—is crucial to the efficiency of these methods. Rather than specify it explicitly, the VGP adaptively learns this transformation and avoids discretization.

Limiting the VGP in various ways recovers well-known probability models as variational approximations. Specifically, we recover the discrete mixture of mean-field distributions (Bishop et al., 1998; Jaakkola & Jordan, 1998). We also recover a form of factor analysis (Tipping & Bishop, 1999) in the variational space. Mathematical details are in Appendix A.

5 Experiments

Following standard benchmarks for variational inference in deep learning, we learn generative models of images. In particular, we learn the deep latent Gaussian model (DLGM) (Rezende et al., 2014), a layered hierarchy of Gaussian random variables following neural network architecures, and the recently proposed Deep Recurrent Attentive Writer (DRAW) (Gregor et al., 2015), a latent attention model that iteratively constructs complex images using a recurrent architecture and a sequence of variational auto-encoders (Kingma & Welling, 2014).

For the learning rate we apply a version of RMSProp (Tieleman & Hinton, 2012), in which we scale the value with a decaying schedule $1/t^{1/2+\epsilon}$ for $\epsilon > 0$. We fix the size of variational data to be 500 across all experiments and set the latent input dimension equal to the number of latent variables.

5.1 Binarized MNIST

The binarized MNIST data set (Salakhutdinov & Murray, 2008) consists of 28x28 pixel images with binary-valued outcomes. Training a DLGM, we apply two stochastic layers of 100 random variables and 50 random variables respectively, and in-between each stochastic layer is a deterministic

Model	Epochs	$\leq -\log p(\mathbf{x})$
DRAW	100	526.8
	200	479.1
	300	464.5
DRAW + VGP	100	475.9
	200	430.0
	300	425.4

Table 2: Negative predictive log-likelihood for Sketch, learned over hundreds of epochs over all 18,000 training examples.

Figure 3: Generated images from DRAW with a VGP (top), and DRAW with the original variational auto-encoder (bottom). The VGP learns texture and sharpness, able to sketch more complex shapes.

layer with 100 units using tanh nonlinearities. We apply mean-field Gaussian distributions for the stochastic layers and a Bernoulli likelihood. We train the VGP to learn the DLGM for the cases of one stochastic layer and two stochastic layers.

For DRAW (Gregor et al., 2015), we augment the mean-field Gaussian distribution originally used to generate the latent samples at each time step with the VGP, as it places a complex variational prior over its parameters. The encoding recurrent neural network now outputs variational data (used for the variational model) as well as mean-field Gaussian parameters (used for the auxiliary model). We use the same architecture hyperparameters as in Gregor et al. (2015).

After training we evaluate test set log likelihood, which are lower bounds on the true value. See Table 1 which reports both approximations and lower bounds of $\log p(\mathbf{x})$ for various methods. The VGP achieves the highest known results on log-likelihood using DRAW, reporting a value of **-80.11** compared to the original highest of -80.97. The VGP also achieves the highest known results among the class of non-structure exploiting models using the DLGM, with a value of -81.90 compared to the previous best of -82.90 reported by Burda et al. (2016).

5.2 SKETCH

As a demonstration of the VGP's complexity for learning representations, we also examine the Sketch data set (Eitz et al., 2012). It consists of 20,000 human sketches equally distributed over 250 object categories. We partition it into 18,000 training examples and 2,000 test examples. We fix the architecture of DRAW to have a 2x2 read window, 5x5 write attention window, and 64 glimpses—these values were selected using a coarse grid search and choosing the set which lead to the best training log likelihood. For inference we use the original auto-encoder version as well as the augmented version with the VGP.

See Table 2. DRAW with the VGP achieves a significantly better lower bound, performing better than the original version which has seen state-of-the-art success in many computer vision tasks. (Until the results presented here, the results from the original DRAW were the best reported performance for this data set.). Moreover, the model inferred using the VGP is able to generate more complex images than the original version—it not only performs better but maintains higher visual fidelity.

6 DISCUSSION

We present the variational Gaussian process (VGP), a variational model which adapts its shape to match complex posterior distributions. The VGP draws samples from a tractable distribution, and posits a Bayesian nonparametric prior over transformations from the tractable distribution to mean-field parameters. The VGP learns the transformations from the space of all continuous mappings—it is a universal approximator and finds good posterior approximations via optimization.

In future work the VGP will be explored for application in Monte Carlo methods, where it may be an efficient proposal distribution for importance sampling and sequential Monte Carlo. An important avenue of research is also to characterize local optima inherent to the objective function. Such analysis will improve our understanding of the limits of the optimization procedure and thus the limits of variational inference.

ACKNOWLEDGEMENTS

We thank David Duvenaud, Alp Kucukelbir, Ryan Giordano, and the anonymous reviewers for their helpful comments. This work is supported by NSF IIS-0745520, IIS-1247664, IIS-1009542, ONR N00014-11-1-0651, DARPA FA8750-14-2-0009, N66001-15-C-4032, Facebook, Adobe, Amazon, and the Seibel and John Templeton Foundations.

REFERENCES

Agakov, Felix V and Barber, David. An auxiliary variational method. In *Neural Information Processing*, pp. 561–566. Springer, 2004.

Bergstra, James, Breuleux, Olivier, Bastien, Frédéric, Lamblin, Pascal, Pascanu, Razvan, Desjardins, Guillaume, Turian, Joseph, Warde-Farley, David, and Bengio, Yoshua. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, 2010.

Bishop, Christopher M., Lawrence, Neil D., Jordan, Michael I., and Jaakkola, Tommi. Approximating posterior distributions in belief networks using mixtures. In *Neural Information Processing Systems*, 1998.

Burda, Yuri, Grosse, Roger, and Salakhutdinov, Ruslan. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016.

Carpenter, Bob, Hoffman, Matthew D., Brubaker, Marcus, Lee, Daniel, Li, Peter, and Betancourt, Michael. The Stan Math Library: Reverse-mode automatic differentiation in C++. *arXiv preprint arXiv:1509.07164*, 2015.

Chopin, Nicolas. A sequential particle filter method for static models. *Biometrika*, 89(3):539–552, 2002.

Cunningham, John P, Shenoy, Krishna V, and Sahani, Maneesh. Fast Gaussian process methods for point process intensity estimation. In *International Conference on Machine Learning*. ACM, 2008.

Dinh, Laurent, Krueger, David, and Bengio, Yoshua. NICE: Non-linear independent components estimation. In *International Conference on Learning Representations Workshop*, 2015.

Eitz, Mathias, Hays, James, and Alexa, Marc. How do humans sketch objects? *ACM Trans. Graph. (Proc. SIGGRAPH)*, 31(4):44:1–44:10, 2012.

Gelman, Andrew and Meng, Xiao-Li. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, 1998.

Gregor, Karol, Danihelka, Ivo, Mnih, Andriy, Blundell, Charles, and Wierstra, Daan. Deep autoregressive networks. In *International Conference on Machine Learning*, 2014.

Gregor, Karol, Danihelka, Ivo, Graves, Alex, Rezende, Danilo Jimenez, and Wierstra, Daan. DRAW: A recurrent neural network for image generation. In *International Conference on Machine Learning*, 2015.

Hinton, G. and Van Camp, D. Keeping the neural networks simple by minimizing the description length of the weights. In *Computational Learning Theory*, pp. 5–13. ACM, 1993.

Hinton, Geoffrey E and Zemel, Richard S. Autoencoders, minimum description length, and helmholtz free energy. In *Neural Information Processing Systems*, 1994.

Hinton, Geoffrey E, Osindero, Simon, and Teh, Yee-Whye. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

Hoffman, Matthew D, Blei, David M, Wang, Chong, and Paisley, John. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

Jaakkola, Tommi S and Jordan, Michael I. Improving the mean field approximation via the use of mixture distributions. In *Learning in Graphical Models*, pp. 163–173. Springer Netherlands, Dordrecht, 1998.

Jordan, Michael I, Ghahramani, Zoubin, Jaakkola, Tommi S, and Saul, Lawrence K. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

Kingma, Diederik P and Welling, Max. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Kucukelbir, Alp, Tran, Dustin, Ranganath, Rajesh, Gelman, Andrew, and Blei, David M. Automatic differentiation variational inference. *arXiv preprint arXiv:1603.00788*, 2016.

Lawrence, Neil. *Variational Inference in Probabilistic Models*. PhD thesis, 2000.

Lawrence, Neil. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *The Journal of Machine Learning Research*, 6:1783–1816, 2005.

Murray, Iain and Salakhutdinov, Ruslan R. Evaluating probabilities under high-dimensional latent variable models. In *Advances in neural information processing systems*, pp. 1137–1144, 2009.

Neal, Radford M. Learning stochastic feedforward networks. *Department of Computer Science, University of Toronto*, 1990.

Neal, Radford M. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 1998.

Nelsen, Roger B. *An Introduction to Copulas (Springer Series in Statistics)*. Springer-Verlag New York, Inc., 2006.

Osborne, Michael. *Bayesian Gaussian processes for sequential prediction, optimisation and quadrature*. PhD thesis, Oxford University New College, 2010.

Quiñonero-Candela, Joaquin and Rasmussen, Carl Edward. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

Raiko, Tapani, Li, Yao, Cho, Kyunghyun, and Bengio, Yoshua. Iterative neural autoregressive distribution estimator nade-k. In *Advances in Neural Information Processing Systems*, pp. 325–333, 2014.

Ranganath, Rajesh, Gerrish, Sean, and Blei, David M. Black box variational inference. In *Artificial Intelligence and Statistics*, 2014.

Ranganath, Rajesh, Tang, Linpeng, Charlin, Laurent, and Blei, David M. Deep exponential families. In *Artificial Intelligence and Statistics*, 2015a.

Ranganath, Rajesh, Tran, Dustin, and Blei, David M. Hierarchical variational models. *arXiv preprint arXiv:1511.02386*, 2015b.

Rasmussen, Carl Edward and Williams, Christopher K I. *Gaussian processes for machine learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2006.

Rezende, Danilo Jimenez and Mohamed, Shakir. Variational inference with normalizing flows. In *International Conference on Machine Learning*, 2015.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

Robbins, Herbert and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics*, 1951.

Salakhutdinov, Ruslan and Larochelle, Hugo. Efficient learning of deep Boltzmann machines. In *International Conference on Artificial Intelligence and Statistics*, pp. 693–700, 2010.

Salakhutdinov, Ruslan and Murray, Iain. On the quantitative analysis of deep belief networks. In *International Conference on Machine Learning*, 2008.

Salimans, Tim, Kingma, Diederik P, and Welling, Max. Markov chain Monte Carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, 2015.

Saul, Lawrence K and Jordan, Michael I. Exploiting tractable substructures in intractable networks. In *Neural Information Processing Systems*, 1996.

Tieleman, T. and Hinton, G. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

Tipping, Michael E and Bishop, Christopher M. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

Tran, Dustin, Blei, David M., and Airoldi, Edoardo M. Copula variational inference. In *Neural Information Processing Systems*, 2015.

Van Der Vaart, Aad and Van Zanten, Harry. Information rates of nonparametric Gaussian process methods. *The Journal of Machine Learning Research*, 12:2095–2119, 2011.

Wainwright, Martin J and Jordan, Michael I. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

Waterhouse, S., MacKay, D., and Robinson, T. Bayesian methods for mixtures of experts. In *Neural Information Processing Systems*, 1996.

A SPECIAL CASES OF THE VARIATIONAL GAUSSIAN PROCESS

We now analyze two special cases of the VGP: by limiting its generative process in various ways, we recover well-known models. This provides intuition behind the VGP's complexity. In Section 4 we show many recently proposed models can also be viewed as special cases of the VGP.

Special Case 1. *A mixture of mean-field distributions is a* VGP *without a kernel.*

A discrete mixture of mean-field distributions (Bishop et al., 1998; Jaakkola & Jordan, 1998; Lawrence, 2000) is a classically studied variational model with dependencies between latent variables. Instead of a mapping which interpolates between inputs of the variational data, suppose the VGP simply performs nearest-neighbors for a latent input $\boldsymbol{\xi}$—selecting the output t_n tied to the nearest variational input s_n. This induces a multinomial distribution of outputs, which samples one of the variational outputs' mean-field parameters.[2] Thus, with a GP prior that interpolates between inputs, the VGP can be seen as a kernel density smoothing of the nearest-neighbor function.

Special Case 2. *Variational factor analysis is a* VGP *with linear kernel and no variational data.*

Consider factor analysis (Tipping & Bishop, 1999) in the variational space: [3]

$$\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \qquad \mathbf{z}_i \sim \mathcal{N}(\mathbf{w}^\top \boldsymbol{\xi}, \mathbf{I}).$$

Marginalizing over the latent inputs induces linear dependence in $\mathbf{z}$, $q(\mathbf{z}; \mathbf{w}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{w}\mathbf{w}^\top)$. Consider the dual interpretation

$$\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \qquad f_i \sim \mathcal{GP}(0, k(\cdot, \cdot)), k(\mathbf{s}, \mathbf{s}') = \mathbf{s}^\top \mathbf{s}', \qquad \mathbf{z}_i = f_i(\boldsymbol{\xi}),$$

with $q(\mathbf{z} \mid \boldsymbol{\xi}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \boldsymbol{\xi}\boldsymbol{\xi}^\top)$. The maximum likelihood estimate of $\mathbf{w}$ in factor analysis is the maximum a posteriori estimate of $\boldsymbol{\xi}$ in the GP formulation. More generally, use of a non-linear kernel induces non-linear dependence in $\mathbf{z}$. Learning the set of kernel hyperparameters $\boldsymbol{\theta}$ thus learns the set capturing the most variation in its latent embedding of $\mathbf{z}$ (Lawrence, 2005).

B PROOF OF THEOREM 1

Theorem 1. *Let $q(\mathbf{z}; \boldsymbol{\theta}, \mathcal{D})$ denote the variational Gaussian process. Consider a posterior distribution $p(\mathbf{z} \mid \mathbf{x})$ with a finite number of latent variables and continuous quantile function (inverse CDF). There exists a sequence of parameters $(\boldsymbol{\theta}_k, \mathcal{D}_k)$ such that*

$$\lim_{k \to \infty} \mathrm{KL}(q(\mathbf{z}; \boldsymbol{\theta}_k, \mathcal{D}_k) \,\|\, p(\mathbf{z} \mid \mathbf{x})) = 0.$$

[2] Formally, given variational input-output pairs $\{(\mathbf{s}_n, \mathbf{t}_n)\}$, the nearest-neighbor function is defined as $f(\boldsymbol{\xi}) = \mathbf{t}_j$, such that $\|\boldsymbol{\xi} - \mathbf{s}_j\| < \|\boldsymbol{\xi} - \mathbf{s}_k\|$ for all k. Then the output's distribution is multinomial with probabilities $P(f(\boldsymbol{\xi}) = \mathbf{t}_j)$, proportional to areas of the partitioned nearest-neighbor space.

[3] For simplicity, we avoid discussion of the VGP's underlying mean-field distribution, i.e., we specify each mean-field factor to be a degenerate point mass at its parameter value.

Proof. Let the mean-field distribution be given by degenerate delta distributions

$$q(\mathbf{z}_i \mid f_i) = \delta_{f_i}(\mathbf{z}_i).$$

Let the size of the latent input be equivalent to the number of latent variables $c = d$ and fix $\sigma^2_{\text{ARD}} = 1$ and $\boldsymbol{\omega}_j = 1$. Furthermore for simplicity, we assume that $\boldsymbol{\xi}$ is drawn uniformly on the d-dimensional hypercube. Then as explained in Section 2.4, if we let P^{-1} denote the inverse posterior cumulative distribution function, the optimal f denoted f^* such that

$$\mathrm{KL}(q(\mathbf{z}; \boldsymbol{\theta}) \,\|\, p(\mathbf{z} \mid \mathbf{x})) = 0$$

is

$$f^*(\xi) = P^{-1}(\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_d).$$

Define $\mathcal{O}_k$ to be the set of points $j/2^k$ for $j = 0$ to 2^k, and define $\mathcal{S}_k$ to be the d-dimensional product of $\mathcal{O}_k$. Let $\mathcal{D}_k$ be the set containing the pairs $(s_i, f^*(s_i))$, for each element s_i in $\mathcal{S}_k$. Denote f^k as the GP mapping conditioned on the dataset $\mathcal{D}_k$, this random mapping satisfies $f^k(s_i) = f^*(s_i)$ for all $s_i \in \mathcal{S}_k$ by the noise free prediction property of Gaussian processes (Rasmussen & Williams, 2006). Then by continuity, as $k \to \infty$, f^k converges to f^*. □

A broad condition under which the quantile function of a distribution is continuous is if that distribution has positive density with respect to the Lebesgue measure.

The rate of convergence for finite sizes of the variational data can be studied via posterior contraction rates for GPs under random covariates (Van Der Vaart & Van Zanten, 2011). Only an additional assumption using stronger continuity conditions for the posterior quantile and the use of Matern covariance functions is required for the theory to be applicable in the variational setting.

C VARIATIONAL OBJECTIVE

We derive the tractable lower bound to the model evidence $\log p(\mathbf{x})$ presented in Eq.6. To do this, we first penalize the ELBO with an expected KL term,

$$\begin{aligned}\log p(\mathbf{x}) &\geq \mathcal{L} = \mathbb{E}_{q_{\text{VGP}}}[\log p(\mathbf{x} \mid \mathbf{z})] - \mathrm{KL}(q_{\text{VGP}}(\mathbf{z}) \| p(\mathbf{z})) \\ &\geq \mathbb{E}_{q_{\text{VGP}}}[\log p(\mathbf{x} \mid \mathbf{z})] - \mathrm{KL}(q_{\text{VGP}}(\mathbf{z}) \| p(\mathbf{z})) - \mathbb{E}_{q_{\text{VGP}}}\Big[\mathrm{KL}(q(\boldsymbol{\xi}, f \mid \mathbf{z}) \| r(\boldsymbol{\xi}, f \mid \mathbf{z}))\Big].\end{aligned}$$

We can combine all terms into the expectations as follows:

$$\begin{aligned}\widetilde{\mathcal{L}} &= \mathbb{E}_{q(\mathbf{z}, \boldsymbol{\xi}, f)}\Big[\log p(\mathbf{x} \mid \mathbf{z}) - \log q(\mathbf{z}) + \log p(\mathbf{z}) - \log q(\boldsymbol{\xi}, f \mid \mathbf{z}) + \log r(\boldsymbol{\xi}, f \mid \mathbf{z})\Big] \\ &= \mathbb{E}_{q(\mathbf{z}, \boldsymbol{\xi}, f)}\Big[\log p(\mathbf{x} \mid \mathbf{z}) - \log q(\mathbf{z} \mid f(\boldsymbol{\xi})) + \log p(\mathbf{z}) - \log q(\boldsymbol{\xi}, f) + \log r(\boldsymbol{\xi}, f \mid \mathbf{z})\Big],\end{aligned}$$

where we apply the product rule $q(\mathbf{z})q(\boldsymbol{\xi}, f \mid \mathbf{z}) = q(\mathbf{z} \mid f(\boldsymbol{\xi}))q(\boldsymbol{\xi}, f)$. Recombining terms as KL divergences, and written with parameters $(\boldsymbol{\theta}, \boldsymbol{\phi})$, this recovers the auto-encoded variational objective in Section 3:

$$\begin{aligned}\widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{q_{\text{VGP}}}[\log p(\mathbf{x} \mid \mathbf{z})] &- \mathbb{E}_{q_{\text{VGP}}}\Big[\mathrm{KL}(q(\mathbf{z} \mid f(\boldsymbol{\xi})) \| p(\mathbf{z}))\Big] \\ &- \mathbb{E}_{q_{\text{VGP}}}\Big[\mathrm{KL}(q(f \mid \boldsymbol{\xi}; \boldsymbol{\theta}) \| r(f \mid \boldsymbol{\xi}, \mathbf{z}; \boldsymbol{\phi})) + \log q(\boldsymbol{\xi}) - \log r(\boldsymbol{\xi} \mid \mathbf{z})\Big].\end{aligned}$$

The KL divergence between the mean-field $q(\mathbf{z} \mid f(\boldsymbol{\xi}))$ and the model prior $p(\mathbf{z})$ is analytically tractable for certain popular models. For example, in the deep latent Gaussian model (Rezende et al., 2014) and DRAW (Gregor et al., 2015), both the mean-field distribution and model prior are Gaussian, leading to an analytic KL term: for Gaussian random variables of dimension d,

$$\begin{aligned}&\mathrm{KL}(\mathcal{N}(\mathbf{x}; \mathbf{m}_1, \boldsymbol{\Sigma}_1) \| \mathcal{N}(\mathbf{x}; \mathbf{m}_2, \boldsymbol{\Sigma}_2)) = \\ &\quad \frac{1}{2}\left((\mathbf{m}_1 - \mathbf{m}_2)^\top \boldsymbol{\Sigma}_1^{-1}(\mathbf{m}_1 - \mathbf{m}_2) + \mathrm{tr}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_2 + \log \boldsymbol{\Sigma}_1 - \log \boldsymbol{\Sigma}_2) - d\right).\end{aligned}$$

In general, when the KL is intractable, we combine the KL term with the reconstruction term, and maximize the variational objective

$$\begin{aligned}\widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}) &= \mathbb{E}_{q_{\text{VGP}}}[\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z} \mid f(\boldsymbol{\xi}))] \\ &\quad - \mathbb{E}_{q_{\text{VGP}}}\Big[\operatorname{KL}(q(f \mid \boldsymbol{\xi}; \boldsymbol{\theta}) \| r(f \mid \boldsymbol{\xi}, \mathbf{z}; \boldsymbol{\phi})) + \log q(\boldsymbol{\xi}) - \log r(\boldsymbol{\xi} \mid \mathbf{z})\Big].\end{aligned} \tag{8}$$

We expect that this experiences slightly higher variance in the stochastic gradients during optimization.

We now consider the second term. Recall that we specify the auxiliary model to be a fully factorized Gaussian, $r(\boldsymbol{\xi}, f \mid \mathbf{z}) = \mathcal{N}((\boldsymbol{\xi}, f(\boldsymbol{\xi}))^\top \mid \mathbf{z}; \mathbf{m}, \mathbf{S})$, where $\mathbf{m} \in \mathbb{R}^{c+d}$, $\mathbf{S} \in \mathbb{R}^{c+d}$. Further, the variational priors $q(\boldsymbol{\xi})$ and $q(f \mid \boldsymbol{\xi})$ are both defined to be Gaussian. Therefore it is also a KL divergence between Gaussian distributed random variables. Similarly, $\log q(\boldsymbol{\xi}) - \log r(\boldsymbol{\xi} \mid \mathbf{z})$ is simply a difference of Gaussian log densities. The second expression is simple to compute and backpropagate gradients.

D GRADIENTS OF THE VARIATIONAL OBJECTIVE

We derive gradients for the variational objective (Eq.7). This follows trivially by backpropagation:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} \widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}) &= \mathbb{E}_{\mathcal{N}(\boldsymbol{\xi})}[\mathbb{E}_{w(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}} \mathbf{f}(\boldsymbol{\xi}) \nabla_{\mathbf{f}} \mathbf{z}(\boldsymbol{\epsilon}) \nabla_{\mathbf{z}} \log p(\mathbf{x} \mid \mathbf{z})]] \\ &\quad - \mathbb{E}_{\mathcal{N}(\boldsymbol{\xi})}\Big[\mathbb{E}_{w(\boldsymbol{\epsilon})}\Big[\nabla_{\boldsymbol{\theta}} \operatorname{KL}(q(\mathbf{z} \mid \mathbf{f}(\boldsymbol{\xi}; \boldsymbol{\theta})) \| p(\mathbf{z}))\Big]\Big] \\ &\quad - \mathbb{E}_{\mathcal{N}(\boldsymbol{\xi})}\Big[\mathbb{E}_{w(\boldsymbol{\epsilon})}\Big[\nabla_{\boldsymbol{\theta}} \operatorname{KL}(q(f \mid \boldsymbol{\xi}; \boldsymbol{\theta}) \| r(f \mid \boldsymbol{\xi}, \mathbf{z}; \boldsymbol{\phi}))\Big]\Big], \\ \nabla_{\boldsymbol{\phi}} \widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}) &= -\mathbb{E}_{\mathcal{N}(\boldsymbol{\xi})}[\mathbb{E}_{w(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\phi}} \operatorname{KL}(q(f \mid \boldsymbol{\xi}; \boldsymbol{\theta}) \| r(f \mid \boldsymbol{\xi}, \mathbf{z}; \boldsymbol{\phi})) - \nabla_{\boldsymbol{\phi}} \log r(\boldsymbol{\xi} \mid \mathbf{z}; \boldsymbol{\phi})]],\end{aligned}$$

where we assume the KL terms are analytically written from Appendix C and gradients are propagated similarly through their computational graph. In practice, we need only be careful about the expectations, and the gradients of the functions written above are taken care of with automatic differentiation tools.

We also derive gradients for the general variational bound of Eq.8—it assumes that the first KL term, measuring the divergence between q and the prior for p, is not necessarily tractable. Following the reparameterizations described in Section 3.3, this variational objective can be rewritten as

$$\begin{aligned}\widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}) &= \mathbb{E}_{\mathcal{N}(\boldsymbol{\xi})}\Big[\mathbb{E}_{w(\boldsymbol{\epsilon})}\Big[\log p(\mathbf{x}, \mathbf{z}(\boldsymbol{\epsilon}; \mathbf{f})) - \log q(\mathbf{z}(\boldsymbol{\epsilon}; \mathbf{f}) \mid \mathbf{f})\Big]\Big] \\ &\quad - \mathbb{E}_{\mathcal{N}(\boldsymbol{\xi})}\Big[\mathbb{E}_{w(\boldsymbol{\epsilon})}\Big[\operatorname{KL}(q(f \mid \boldsymbol{\xi}; \boldsymbol{\theta}) \| r(f \mid \boldsymbol{\xi}, \mathbf{z}(\boldsymbol{\epsilon}; \mathbf{f}); \boldsymbol{\phi})) + \log q(\boldsymbol{\xi}) - \log r(\boldsymbol{\xi} \mid \mathbf{z}(\boldsymbol{\epsilon}; \mathbf{f}))\Big]\Big].\end{aligned}$$

We calculate gradients by backpropagating over the nested reparameterizations:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} \widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}) &= \mathbb{E}_{\mathcal{N}(\boldsymbol{\xi})}[\mathbb{E}_{w(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}} \mathbf{f}(\boldsymbol{\xi}) \nabla_{\mathbf{f}} \mathbf{z}(\boldsymbol{\epsilon})[\nabla_{\mathbf{z}} \log p(\mathbf{x}, \mathbf{z}) - \nabla_{\mathbf{z}} \log q(\mathbf{z} \mid \mathbf{f})]]] \\ &\quad - \mathbb{E}_{\mathcal{N}(\boldsymbol{\xi})}\Big[\mathbb{E}_{w(\boldsymbol{\epsilon})}\Big[\nabla_{\boldsymbol{\theta}} \operatorname{KL}(q(f \mid \boldsymbol{\xi}; \boldsymbol{\theta}) \| r(f \mid \boldsymbol{\xi}, \mathbf{z}; \boldsymbol{\phi}))\Big]\Big] \\ \nabla_{\boldsymbol{\phi}} \widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}) &= -\mathbb{E}_{\mathcal{N}(\boldsymbol{\xi})}[\mathbb{E}_{w(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\phi}} \operatorname{KL}(q(f \mid \boldsymbol{\xi}; \boldsymbol{\theta}) \| r(f \mid \boldsymbol{\xi}, \mathbf{z}; \boldsymbol{\phi})) - \nabla_{\boldsymbol{\phi}} \log r(\boldsymbol{\xi} \mid \mathbf{z}; \boldsymbol{\phi})]].\end{aligned}$$

E SCALING THE SIZE OF VARIATIONAL DATA

If massive sizes of variational data are required, e.g., when its cubic complexity due to inversion of a $m \times m$ matrix becomes the bottleneck during computation, we can scale it further. Consider fixing the variational inputs to lie on a grid. For stationary kernels, this allows us to exploit Toeplitz structure for fast $m \times m$ matrix inversion. In particular, one can embed the Toeplitz matrix into a circulant matrix and apply conjugate gradient combined with fast Fourier transforms in order to compute inverse-matrix vector products in $\mathcal{O}(m \log m)$ computation and $\mathcal{O}(m)$ storage (Cunningham et al., 2008). For product kernels, we can further exploit Kronecker structure to allow fast $m \times m$ matrix inversion in $\mathcal{O}(Pm^{1+1/P})$ operations and $\mathcal{O}(Pm^{2/P})$ storage, where $P > 1$ is the number of kernel products (Osborne, 2010). The ARD kernel specifically leads to $\mathcal{O}(cm^{1+1/c})$ complexity, which is linear in m.