Variational Gaussian Process

Dustin Tran
Harvard University
dtran@g.harvard.edu

Rajesh Ranganath
Princeton University
rajeshr@cs.princeton.edu

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

Abstract

Representations offered by deep generative models are fundamentally tied to their inference method from data. Variational inference methods require a rich family of approximating distributions. We construct the variational Gaussian process (VGP), a Bayesian nonparametric model 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

Originally developed in the 1990s (Hinton and 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 and Welling, 2014; Rezende et al., 2014). This research has been particularly successful for computing with deep Bayesian models (Neal, 1990; Hinton et al., 2006), which require inference of a complex posterior distribution.

Despite these advances, research has been constrained by the lack of an expressive family of approximating distributions, which can generalize across many models. Newer research aims toward richer families that allow the latent variables to be dependent. One way to introduce dependence is to consider the variational family itself as a model of the hidden variables (Lawrence, 2000; Salimans et al., 2015; Ranganath et al., 2015), and then to use models that go beyond the simple mean field. 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 present a variational lower bound inspired by auto-encoders and, for black box inference, derive an efficient stochastic optimization algorithm. We report new state-of-the-art results on the binarized MNIST data set.

2 Variational model

Let $p(\boldsymbol{z} \mid \boldsymbol{x})$ denote a posterior distribution over d latent variables $\boldsymbol{z} = (z_1, \ldots, z_i)$ conditioned on a data set $\boldsymbol{x}$. Assume a choice of mean-field distribution $\prod_{i=1}^{d} q(z_i; \boldsymbol{\lambda}_i)$. For example, each factor can simply be a Gaussian with parameters $\boldsymbol{\lambda}_i = (\mu_i, \sigma_i^2)$. The VGP generates $\boldsymbol{z}$ 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 is itself learned as part of variational inference. We will show that the VGP enables samples from the mean-field to follow arbitrarily complex posteriors.

Let $\mathcal{D} = \{(\mathbf{s}_n, \mathbf{t}_n)\}_{n=1}^m$ be variational data, comprising input-output pairs that are parameters to the variational distribution. (This idea appears in a different context in Blei and Lafferty (2006).) The VGP specifies the following generative process for posterior latent variables $\boldsymbol{z}$:

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 $\mathcal{D}$: $f \sim \prod_{i=1}^d \mathcal{GP}(\mathbf{0}, \mathbf{K}_{\boldsymbol{\xi}\boldsymbol{\xi}}) \mid \mathcal{D}$.
3. Draw approximate posterior samples $\boldsymbol{z} \in \text{supp}(p)$: $\boldsymbol{z} = (z_1, \ldots, z_d) \sim \prod_{i=1}^d q(f_i(\boldsymbol{\xi}))$.

Figure 1 displays a graphical model for the VGP. Marginalizing over all non-linear mappings and latent inputs, the VGP is

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

which 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 specified *conditional* on a fixed function $f(\cdot)$ and input $\boldsymbol{\xi}$; the d outputs $f_i(\boldsymbol{\xi}) = \boldsymbol{\lambda}_i$ are the mean-field's parameters. The VGP is a form of a hierarchical variational model (Ranganath et al., 2015); it places a continuous Bayesian nonparametric prior over mean-field parameters.

We emphasize that the VGP needs variational data because—unlike typical Gaussian process (GP) regression—there is no observed data available to learn a distribution over non-linear mappings. The variational data appear in the conditional distribution of f, anchoring the random non-linear mappings at certain input-ouput pairs. Thus, when we optimize the VGP, the learned variational data enables a complex distribution of variational parameters $f(\boldsymbol{\xi})$.

We show that the VGP is a universal approximator.

Theorem 1 (Universal approximation)**.** *Let $q(\boldsymbol{z}; \boldsymbol{\theta}, \mathcal{D})$ denote the variational Gaussian process. For any posterior distribution $p(\boldsymbol{z} \mid \boldsymbol{x})$ with a finite number of latent variables and continuous quantile function (inverse CDF), there exist a set of parameters $(\boldsymbol{\theta}, \mathcal{D})$ such that*

$$\text{KL}(q(\boldsymbol{z}; \boldsymbol{\theta}, \mathcal{D}) \parallel p(\boldsymbol{z} \mid \boldsymbol{x})) = 0.$$

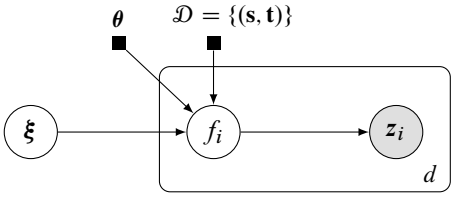
Theorem 1 states that any posterior distribution with strictly positive density can be represented by a VGP. Thus the VGP is a flexible model for learning posterior distributions.

3 Black box inference

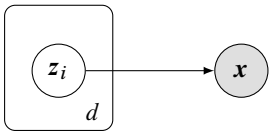
We derive an algorithm for performing black box inference over a wide class of generative models. The original evidence lower bound (ELBO) is analytically intractable due to the log density $\log q_{\text{VGP}}(\boldsymbol{z})$ (Eq.1). Inference on variational models with latent variables typically requires computation of an auxiliary model $r(\boldsymbol{\xi}, f \mid \boldsymbol{z})$ (Salimans et al., 2015; Ranganath et al., 2015). Unlike previous approaches, we present the variational objective in terms of auto-encoders:

$$\widetilde{\mathcal{L}} = \mathbb{E}_{q_{\text{VGP}}}[\log p(\boldsymbol{x} \mid \boldsymbol{z})] - \mathbb{E}_{q_{\text{VGP}}}\Big[\text{KL}(q(\boldsymbol{z} \mid f(\boldsymbol{\xi})) \| p(\boldsymbol{z})) + \text{KL}(q(\boldsymbol{\xi}, f) \| r(\boldsymbol{\xi}, f \mid \boldsymbol{z}))\Big].$$

In auto-encoder parlance, we maximize the expected negative reconstruction error, regularized by 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 and Welling, 2014): a KL 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 the use of several analytic expectations.



(a) VARIATIONAL MODEL



(b) GENERATIVE MODEL

Figure 1: **(a)** Graphical model of the variational Gaussian process. Approximate posterior samples $\boldsymbol{z}$ for the posterior distribution of a generative model **(b)**.

Algorithm 1: Black box inference with a VGP

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

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

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

end

3.1 Auto-encoding variational models

Inference networks provide a flexible parameterization of approximating distributions as used in Helmholtz machines (Hinton and Zemel, 1994), deep Boltzmann machines (Salakhutdinov and Larochelle, 2010), and variational auto-encoders (Kingma and Welling, 2014; Rezende et al., 2014). To auto-encode the VGP we specify inference networks to parameterize both the variational and auxiliary models:

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

where q has local variational parameters given by the variational data $\mathcal{D}_n$, and 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]

3.2 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 standard procedure in black box methods, we write the gradient as an expectation and apply stochastic approximations (Robbins and Monro, 1951), sampling from the variational model and evaluating stochastic gradients.

First, we simply the expression for the stochastic gradients by analytically deriving any tractable expectations. The KL divergence between $r(\boldsymbol{\xi}, f \mid \boldsymbol{z})$ and $q(\boldsymbol{\xi}, f)$ is analytic as we've specified both joint distributions to be Gaussian. The KL divergence between $q(\boldsymbol{z} \mid f(\boldsymbol{\xi}))$ and $p(\boldsymbol{z})$ is standard and used to reduce variance in traditional variational auto-encoders: it is analytic for widely used deep generative models such as the deep latent Gaussian model (Rezende et al., 2014) and deep recurrent attentive writer (Gregor et al., 2015). See Appendix B for these calculations.

To derive black box gradients, we can first reparameterize the VGP, separating noise generation of samples from the parameters in its generative process (Kingma and 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 $q(\boldsymbol{z} \mid f(\boldsymbol{\xi}))$ is also reparameterizable, and let $\boldsymbol{\epsilon} \sim w$ such that $\boldsymbol{z}(\boldsymbol{\epsilon}; \mathbf{f}) \sim q(\boldsymbol{z} \mid f(\boldsymbol{\xi}))$. This two-level reparameterization is equivalent to the generative process for $\boldsymbol{z}$ outlined in Section 2.

We now rewrite the variational objective as

$$\widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{\mathcal{N}(\boldsymbol{\xi})}\Big[\mathbb{E}_{w(\boldsymbol{\epsilon})}\Big[\log p(\boldsymbol{x} \mid \boldsymbol{z}(\boldsymbol{\epsilon}; \mathbf{f}(\boldsymbol{\xi}; \boldsymbol{\theta})))\Big]\Big] \tag{2}$$

[1] We let the kernel hyperparameters of the VGP be fixed across data points.

Model	$-\log p(\boldsymbol{x})$	$\leq$
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., 2015), [2] (Salimans et al., 2015), [3] (Rezende and Mohamed, 2015), [4] (Raiko et al., 2014), [5] (Murray and Salakhutdinov, 2009), [6] (Gregor et al., 2014), [7] (Gregor et al., 2015).

$$-\mathbb{E}_{\mathcal{N}(\boldsymbol{\xi})}\Big[\mathbb{E}_{w(\boldsymbol{\epsilon})}\Big[\mathrm{KL}(q(\boldsymbol{z}\mid\mathbf{f}(\boldsymbol{\xi};\boldsymbol{\theta}))\|p(\boldsymbol{z}))+\mathrm{KL}(q(\boldsymbol{\xi},f;\boldsymbol{\theta})\|r(\boldsymbol{\xi},f\mid\boldsymbol{z}(\boldsymbol{\epsilon};\mathbf{f}(\boldsymbol{\xi};\boldsymbol{\theta}));\boldsymbol{\phi}))\Big]\Big].$$

Eq.2 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.

An outline is given in Algorithm 1. 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 $\boldsymbol{z}$, we apply the score function estimator for gradients of mean-field expectations (Ranganath et al., 2014). Complexity analysis is available in Appendix D.

4 Experiments

The binarized MNIST data set (Salakhutdinov and Murray, 2008) consists of 28x28 pixel images with binary-valued outcomes. Training a deep latent Gaussian model (DLGM), we apply two stochastic layers of 100 random variables and 50 random variables respectively, and inbetween each stochastic layer is a deterministic layer with 100 units using tanh nonlinearities. We apply mean-field Gaussian distributions for the stochastic layers and a Bernoulli likelihood. 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. 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 $\log p(\boldsymbol{x})$ 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.[2] 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. (2015).

5 Conclusion

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 adaptively learns the transformations from the space of all continuous mappings—it is a universal approximator and achieves powerful flexibility in practice.

[2]While superceding DRAW's previous best results, we have not fully explored initialization in the VGP for learning the model. We hypothesize the VGP may achieve better results given more tuning and computation.

References

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

David M Blei and John D Lafferty. Dynamic topic models. In *International Conference on Machine Learning*, 2006.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

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

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

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

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

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

Geoffrey E Hinton and Richard S Zemel. Autoencoders, minimum description length, and helmholtz free energy. *Advances in neural information processing systems*, pages 3–3, 1994.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

S. Waterhouse, D. MacKay, and T. Robinson. Bayesian methods for mixtures of experts. *Neural Information Processing Systems*, pages 351–357, 1996.

A Proof of Theorem 1

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

$$\mathrm{KL}(q(\boldsymbol{z}; \boldsymbol{\theta}, \mathcal{D}) \parallel p(\boldsymbol{z} \mid \boldsymbol{x})) = 0.$$

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

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

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 if we let P^{-1} denote the inverse posterior cumulative distribution function, the optimal f denoted f^* such that

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

is

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

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

B Variational objective

We derive the tractable lower bound to the model evidence $\log p(\boldsymbol{x})$ presented. We first penalize the ELBO with an expected KL term:

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

We can combine all terms into the expectations as follows:

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

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

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

We now analyze the KL terms. Recall that we specify the auxiliary model $r(\boldsymbol{\xi}, f \mid \boldsymbol{z}) = \mathcal{N}((\boldsymbol{\xi}, f(\boldsymbol{\xi}))^\top \mid \boldsymbol{z}; \mathbf{m}, \mathbf{S})$, where $\mathbf{m} \in \mathbb{R}^{c+d}$, $\mathbf{S} \in \mathbb{R}^{c+d}$. The variational prior for the VGP is also jointly Gaussian:

$$q(\boldsymbol{\xi}, f) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\xi} \\ f(\boldsymbol{\xi})\end{bmatrix}; \begin{bmatrix}\mathbf{0} \\ \widetilde{\mathbf{m}}\end{bmatrix}, \begin{bmatrix}\mathbf{I} & \mathbf{C} \\ \mathbf{C}^\top & \widetilde{\mathbf{K}}\end{bmatrix}\right),$$

where the vector $\widetilde{\mathbf{m}}$ has elements $\widetilde{\mathbf{m}}_i = \mathbf{K}_{\boldsymbol{\xi}s}\mathbf{K}_{ss}^{-1}\mathbf{t}_i$, and the matrix $\widetilde{\mathbf{K}}$ is diagonal with elements $\widetilde{\mathbf{K}}_{ii} = \mathbf{K}_{\boldsymbol{\xi}\boldsymbol{\xi}} - \mathbf{K}_{\boldsymbol{\xi}s}\mathbf{K}_{ss}^{-1}\mathbf{K}_{\boldsymbol{\xi}s}^\top$. Applying the law of total expectations, the cross-correlation is

$$\mathbf{C} = \mathbb{E}[\boldsymbol{\xi} f(\boldsymbol{\xi})^\top] - \mathbb{E}[\boldsymbol{\xi}]\mathbb{E}[f(\boldsymbol{\xi})]^\top = \mathbb{E}[\boldsymbol{\xi}\mathbb{E}_{f \mid \boldsymbol{\xi}}[f(\boldsymbol{\xi}) \mid \boldsymbol{\xi}]^\top] = \mathbb{E}\left[\boldsymbol{\xi}\Big[\prod_{i=1}^{d}\mathbf{K}_{\boldsymbol{\xi}s}\mathbf{K}_{ss}^{-1}\mathbf{t}_i\Big]^\top\right].$$

This expectation is analytically intractable as $\boldsymbol{\xi}$ appears non-linearly in the kernel. In the special case when the kernel is linear, the expression is analytic and corresponds to probabilistic principal components analysis (Lawrence, 2005). In general we apply a Monte Carlo estimate using standard normal samples of $\boldsymbol{\xi}$; for black box gradients we obtain these samples for free as they already follow the process of sampling $\boldsymbol{z}$ (producing intermediate samples $\boldsymbol{\xi}$).

Let $\mathbf{m}_q, \boldsymbol{\Sigma}_q$ denote the concatenated mean vector and covariance matrix for $q(\boldsymbol{\xi}, f)$. Because $q(\boldsymbol{\xi}, f)$ and $r(\boldsymbol{\xi}, f \mid \boldsymbol{z})$ are both Gaussian, the KL has an analytic form:

$$\begin{aligned}&\mathrm{KL}(q(\boldsymbol{\xi}, f; \mathbf{m}_q, \boldsymbol{\Sigma}_q) \| r(\boldsymbol{\xi}, f \mid \boldsymbol{z}; \mathbf{m}, \mathbf{S})) = \\ &\quad\frac{1}{2}\left((\mathbf{m} - \mathbf{m}_q)^\top\mathbf{S}^{-1}(\mathbf{m} - \mathbf{m}_q) + \mathrm{tr}(\mathbf{S}^{-1}\boldsymbol{\Sigma}_q + \log\mathbf{S} - \log\boldsymbol{\Sigma}_q) - (c + d)\right).\end{aligned}$$

Since $\mathbf{S}$ is a diagonal matrix, inversion is trivial—the whole expression is simple to compute and backpropagate gradients.

We now consider the first KL term. The KL divergence between the mean-field $q(\boldsymbol{z} \mid f(\boldsymbol{\xi}))$ and the model prior $p(\boldsymbol{z})$ 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 similar to the above. In general, when the KL is intractable, we combine the term with the reconstruction term, and maximize the variational objective

$$\widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{q(\boldsymbol{z},\boldsymbol{\xi},f;\boldsymbol{\theta})}[\log p(\boldsymbol{x}, \boldsymbol{z}) - q(\boldsymbol{z} \mid f(\boldsymbol{\xi}))] - \mathbb{E}_{q(\boldsymbol{z};\boldsymbol{\theta})}[\mathrm{KL}(q(\boldsymbol{\xi}, f; \boldsymbol{\theta}) \| r(\boldsymbol{\xi}, f \mid \boldsymbol{z}; \boldsymbol{\phi}))], \quad (3)$$

which leads to slightly higher variance in the stochastic gradients during optimization.

C Gradients of the variational objective

We derive gradients for the variational objective (Eq.2). This follows trivially by backpropagation:

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

where we assume the KL terms are analytically written from Appendix B and gradients are propagated similarly through their computational graph.

We also derive gradients for the general variational bound of Eq.3—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 3.2, this variational objective can be rewritten as

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

We calculate gradients by backpropagating over the nested reparameterizations:

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

D Computational and storage complexity

The algorithm has $\mathcal{O}(d + m^3 + LN^2)$ 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 N the average hidden layer size. In particular, the algorithm is linear in the number of latent variables, which is competitive with other variational methods. The number of variational and auxiliary parameters has $\mathcal{O}(c + LN)$ complexity—storing the kernel hyperparameters and the neural network parameters. Note that unlike most GP literature, we require no low rank constraints such as the use of inducing variables (Quiñonero-Candela and Rasmussen, 2005) for scalable computation.

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 automatic relevance determination (ARD) kernel specifically leads to $\mathcal{O}(cm^{1+1/c})$ complexity, which is linear in m.