JMLR: Workshop and Conference Proceedings 44 (2015) 145-159 NIPS 2015

The 1st International Workshop "Feature Extraction: Modern Questions and Challenges"

# The Deep Feed-Forward Gaussian Process: An Effective Generalization to Covariance Priors

**Melih Kandemir and Fred A. Hamprecht**
**Heidelberg University, HCI/IWR**

**Editor:** Afshin Rostamizadeh

## Abstract

We explore ways of applying a prior on the covariance matrix of a Gaussian Process (GP) in order to increase its expressive power. We show that two well-known covariance priors, Wishart Process and Inverse Wishart Process, boil down to a two-layer feed-forward network of GPs with a particular kernel function on the neuron at the output layer. Both of these models perform supervised manifold learning and target prediction jointly. Also, the resultant kernel functions of both of these priors lead to feature maps of finite dimensionality. Motivated by this fact, we promote replacing these kernels with the Radial Basis Function (RBF), which gives an infinite dimensional feature map, enhancing the model flexibility. We demonstrate on one benchmark task and two challenging medical image analysis tasks that our GP network with RBF kernel largely outperforms the earlier two covariance priors. We show also that it straightforwardly allows non-linear combination of different data views, leading to state-of-the-art multiple kernel learning only as a by-product.

## 1. Introduction

Gaussian processes (GPs) attract wide interest as generic supervised learners. This is not only due to their high expressive power coming from their kernelized nature, but also their theoretical connections to a wide spectrum of other methods. For instance, the probabilistic counterpart of the Support Vector Machine (SVM), the Relevance Vector Machine (RVM) is equivalent to a GP with a certain kernel function (Rasmussen and Williams, 2006). Due to their probabilistic nature, GPs can be plugged into a larger probabilistic model as a component. Unlike discriminative models, they take into account the variance of the predicted data points, which is shown to boost up the prediction performance (Seeger, 2003). This probabilistic nature and the predictive variance has also been used to develop simple, effective, and theoretically well-grounded active learning models (Houlsby et al., 2012; Kandemir et al., 2015). GPs allow a principled scheme for learning kernel hyperparameters, for which a grid search is required in the SVM framework.

There exist a number of alternative ways of placing a prior over the covariance matrix of a GP in order to increase its expressive power. Two most intuitive options are the two conjugate priors to the covariance matrix of the normal distribution: i) the Wishart Process (WP) (Zhang et al., 2006), and ii) the Inverse Wishart Process (IWP) (Shah et al., 2014). The WP has been used earlier for relational learning with success (Li et al., 2009), where the WP is placed as a kernel prior, which is then linked to a relational likelihood. Later on, the WP has been generalized by Wilson and Ghahramani (2011) into a prior over *groups*

of matrices evolving in time, and has been used to model the trends of multiple financial metrics. On the other hand, Shah et al. (2014) recently found out that placing the IWP process as a prior on GP covariance increases model flexibility. When the covariance matrix is integrated out, this model leads to the Student-t process.

In this paper, we point attention on the thus far missed fact that either of these two solutions are indeed special cases of a two-layer neural network, where each neuron is a Gaussian process and a particular kernel function is employed on the output neuron. We further show that both of these kernel functions operate on finite dimensional feature maps. Hence, we promote replacing them with a Radial Basis Function (RBF) kernel function, which has an infinite-dimensional feature map, leading to increased expressive power. In the resultant model, GPs on the hidden layer project the input patterns onto a latent manifold non-linearly, and the GP on the output layer maps these projections to the target output. As the posterior distribution of the model is not tractable, we first sparsify all the GPs (i.e. all neurons) and infer the resultant model effectively using variational Bayes. We clarify the connections of this model to Deep GP (Damianou and Lawrence, 2013) and Warped GP (Snelson et al., 2004).

We evaluate our model on one benchmark $SO_2$ level prediction task and two challenging medical image analysis tasks: i) survival time prediction of Barrett's cancer patients from histopathology tissue images, and ii) mitosis and apoptosis detection in time-lapse phase-contrast microscopy image sequences of live human osteosarcoma cell cultures. The outcome of our experiments is that the two-layer DFGP improves the state-of-the-art prediction performance in all three applications, and also that it is able to incorporate heterogeneous input sources and perform multiple kernel learning (MKL) only as a by-product. The source code of our model is publicly available [1].

## 2. Gaussian Processes with Input Priors

Let $\mathbf{X}$ be the $N \times D$ dimensional data matrix with $N$ instances of $D$ dimensions in its rows, and $\mathbf{y}$ be the $N \times 1$ vector of the corresponding outputs. Consider the model below

$$p(\mathbf{B}|\mathbf{X}) = \text{Arbitrary distribution}, \tag{1}$$
$$p(\mathbf{f}|\mathbf{B}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K_{BB}}), \tag{2}$$
$$p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \beta^{-1}\mathbf{I}). \tag{3}$$

Here, Equations 2 and 3 alone amount to the standard GP with fixed input $\mathbf{B}$ and the covariance matrix $\mathbf{K_{BB}}$ governed by a kernel function (i.e. $[\mathbf{K}_{BB}]_{ij} = k(\mathbf{b}_i, \mathbf{b}_j)$). We build on the case that the inputs of the GP are also random variables following the distribution in Equation 1. This setting enriches the GP model family, having many existing models besides the standard GP as its special cases. For instance, Bayesian GPLVM (Titsias and Lawrence, 2010) is the multioutput version of this setup where $p(\mathbf{B}) = \prod_{r=1}^{R} \mathcal{N}(\mathbf{b}_r|\mathbf{0}, \mathbf{I})$. The Deep GP (Damianou and Lawrence, 2013) has another GPLVM as $p(\mathbf{B})$, while the Noisy-input Sparse GP (NSGP) (Pacheco et al., 2014) applies constant white noise to each input $p(\mathbf{B}|\mathbf{X}) = \prod_{n}^{N} \mathcal{N}(\mathbf{b}_n|\mathbf{x}_n, \alpha\mathbf{I})$.

1. `https://github.com/melihkandemir/dfgp`

## 3. The Deep Feed-Forward Gaussian Process

We propose a feed-forward network of GPs, which leads to the following model

$$p(\mathbf{B}|\mathbf{X}) = \prod_{r=1}^{\nu} \mathcal{N}(\mathbf{b}_r|\mathbf{0}, \mathbf{K}_{\mathbf{XX}}), \tag{4}$$

$$p(\mathbf{f}|\mathbf{B}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}_{\mathbf{BB}}), \tag{5}$$

$$p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \beta^{-1}\mathbf{I}). \tag{6}$$

Here, Equation 4 is the hidden layer and consists of $\nu$ independent GPs. This step projects the input patterns onto a $\nu$-dimensional manifold non-linearly. Next, the GP on the output layer (Equation 5) takes these latent projections and maps them to noise-free outputs $\mathbf{f}$. Equation 6 introduces white noise on the output observations. For binary classification, we additionally include $p(\mathbf{t}|\mathbf{y}) = \prod_{n=1}^{N} Bernoulli(t_n|\Phi(y_n))$, where $\Phi(s) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{s} e^{-\frac{1}{2}s^2} ds$ is the probit link function and $\mathbf{t}$ is the vector of output classes $t_n \in \{0, 1\}$. We refer to this model as a *Deep Feed-forward Gaussian Process (DFGP)*.

**Relation to Deep GP and Warped GP.** DFGP borrows the idea of building a network of sparse GPs from the existing Deep GP (Damianou and Lawrence, 2013). As opposed to DFGP's feed-forward network, the Deep GP builds a cascade of GPLVMs, which projects the input pattern into nested manifolds progressively. This is an unsupervised encoder architecture, meant to perform feature learning. Our DFGP, on the other hand, performs supervised manifold learning and target prediction jointly. Figure 1 illustrates the difference. Warped GP (Snelson et al., 2004) is a DFGP with $\nu = 1$ and the GP at the output neuron uses the 1-dimensional projection of the pattern on the latent manifold also as its mean function, unlike the proposed DFGP.

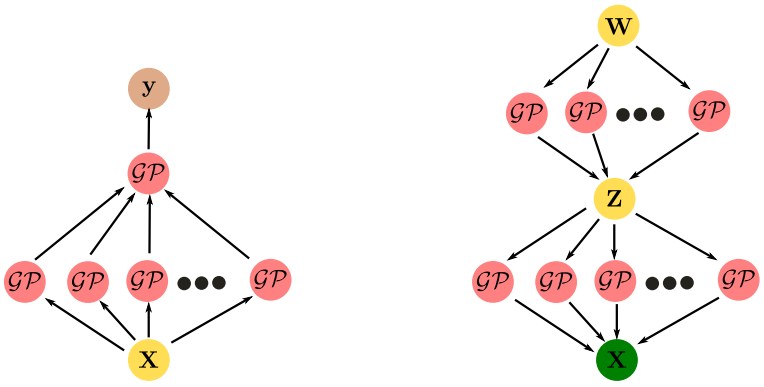



Figure 1: **Left:** A two-layer Deep Feed-Forward Gaussian Process (this work). **Right:** A two layer Deep Gaussian Process (Damianou and Lawrence, 2013).

## 4. DFGP Generalizes WP and IWP Priors

Our central claim is that DFGP leads to a stronger learner than a GP with a WP or IWP prior on its covariance matrix. This is due to that these two latter models boil down to

special cases of DFGP. More specifically, they both are DFGPs with certain kernel functions. An additional reason is that both of these kernel functions lead to finite-dimensional feature maps. However, we prove below that it is possible to achieve infinite-dimensional feature maps by replacing them with the Radial Basis Function (RBF) kernel.

First, we define the Wishart Process (WP) and the Inverse Wishart Process (IWP).

**Definition 1** *A Wishart process (Zhang et al., 2006) $\mathcal{WP}(k(\mathbf{x},\mathbf{x}'),\nu)$ is said to be a stochastic process defined on a sample space $\mathcal{X}$ and characterized by a kernel matrix $k(\mathbf{x},\mathbf{x}')$ and degrees of freedom (DoF) $\nu$ if any finite subset $\{\mathbf{x}_1,\cdots,\mathbf{x}_N\} \in \mathcal{X}$ for any $N$ follows the Wishart distribution $\mathcal{W}(\mathbf{K},\nu)$, where $\mathbf{K}$ is the $N \times N$ Gram matrix with $\mathbf{K}_{ij} = k(\mathbf{x}_i,\mathbf{x}_j)$.*

**Definition 2** *An Inverse Wishart process (Shah et al., 2014) $\mathcal{IWP}(k(\mathbf{x},\mathbf{x}'),\nu)$ is said to be a stochastic process defined on a sample space $\mathcal{X}$ and characterized by a kernel matrix $k(\mathbf{x},\mathbf{x}')$ and degrees of freedom (DoF) $\nu$ if any finite subset $\{\mathbf{x}_1,\cdots,\mathbf{x}_N\} \in \mathcal{X}$ for any $N$ follows the Inverse Wishart distribution $\mathcal{IW}(\mathbf{K},\nu)$, where $\mathbf{K}$ is the $N \times N$ Gram matrix with $\mathbf{K}_{ij} = k(\mathbf{x}_i,\mathbf{x}_j)$.*

Among multiple characterizations of the Inverse Wishart distribution, only the one proposed by Dawid (1981) is consistent over marginalization (i.e. $\Sigma \sim \mathcal{IW}(\mathbf{K},\nu) \Rightarrow \Sigma_{11} \sim \mathcal{IW}(\mathbf{K}_{11},\nu)$ for square matrices $\Sigma_{11}$ and $\mathbf{K}_{11}$ obtained from any same subset of the rows and columns of $\Sigma$ and $\mathbf{K}$, respectively).

Our key contribution is stated by the following theorem.

**Theorem 1.** *Consider the following two models which apply Wishart and Inverse Wishart process priors on the covariance function of a Gaussian process:*

$$\begin{aligned} &i) \quad p(\Sigma) = \mathcal{W}(\Sigma|\mathbf{K},\nu), \quad p(\mathbf{f}|\Sigma) = \mathcal{N}(\mathbf{f}|\mathbf{0},\Sigma), \quad p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{y}|\mathbf{f},\beta^{-1}\mathbf{I});\\ &ii) \quad p(\Sigma) = \mathcal{IW}(\Sigma|\mathbf{K},\nu), \quad p(\mathbf{f}|\Sigma) = \mathcal{N}(\mathbf{f}|\mathbf{0},\Sigma), \quad p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{y}|\mathbf{f},\beta^{-1}\mathbf{I}). \end{aligned}$$

*The Deep Feed-forward Gaussian Process has both (i) and (ii) as its special case.*

**Proof.** *We start with (i). Following the construction process of a Wishart distribution, $\Sigma \sim \mathcal{W}(\Sigma|\mathbf{K},\nu) \equiv \mathbf{B} \sim \prod_{r=1}^{\nu}\mathcal{N}(\mathbf{b}_r|\mathbf{0},\mathbf{K}), \quad \Sigma = \mathbf{B}\mathbf{B}^T$, where $\mathbf{B} = [\mathbf{b}_1 \cdots \mathbf{b}_\nu]$ concatenates the column vectors $\mathbf{b}_r$. The expression on the r.h.s. of the equivalence is $\nu$ independent GPs, corresponding to the hidden layer of the DFGP. Replacing $\Sigma$ by $\mathbf{B}\mathbf{B}^T$ leads to $p(\mathbf{f}|\Sigma) = \mathcal{N}(f|\mathbf{0},\mathbf{B}\mathbf{B}^T)$, which is another GP with the linear kernel function $k(\mathbf{b}_n,\mathbf{b}_{n'}) = \mathbf{b}_n^T\mathbf{b}_{n'}$ for rows $n$ and $n'$ of $\mathbf{B}$, corresponding to the output layer of the DFGP.*

*For (ii), we first use the property that $\Sigma \sim \mathcal{IW}(\Sigma|\mathbf{K},\nu) \equiv \mathbf{V} \sim \mathcal{W}(\mathbf{V}|\mathbf{K},\nu+N-1), \quad \Sigma = \mathbf{V}^{-1}$ (Shah et al., 2014), and the rest follows as in (i). Consequently, we have a DFGP with $\nu + N - 1$ hidden neurons and the kernel function $k(\mathbf{b}_n,\mathbf{b}_{n'}) = \mathbf{E}_{nn'}$, where $\mathbf{E} = [\mathbf{B}\mathbf{B}^T]^{-1}$. Since $\mathbf{B}\mathbf{B}^T$ is positive semi-definite, so is its inverse, hence $\mathbf{E}$ is a valid kernel for the GP at the output neuron.* ■

The DFGP framework allows us to make the above two models (i) and (ii) more flexible, hence more powerful, by increasing the dimensionality of the latent manifold. For this, it suffices to replace the kernel function of the GP at the output neuron with a Radial Basis Function $k(\mathbf{x}_i,\mathbf{x}_j) = \exp(-\frac{1}{2\sigma^2}||\mathbf{x}_i - \mathbf{x}_j||_2^2)$. The following simple corollary summarizes this

outcome.

**Corollary 1.** *The Deep Feed-forward Gaussian Process with an RBF kernel at the output neuron projects the input patterns onto a higher-dimensional manifold than a GP with all possible WP and IWP priors.*

**Proof.** *Trivially, the dimensionality of the manifold resulting from a WP is $\nu$, since for the linear kernel $\phi(\mathbf{b}_n) = \mathbf{b}_n$, where $\phi(\cdot)$ is the feature map. For IWP, the manifold is $\nu + N - 1$-dimensional, as the rank of the resultant inverse covariance is $\nu + N - 1$ and a matrix and its inverse have equal ranks. Since RBF has an infinite dimensional feature map (Smola and Schölkopf, 1998), $\infty > \nu + N - 1 > \nu$ for any $\nu$.* ■

### 4.1 Variational Inference by Sparse GPs

Integrating out the covariance matrix $\Sigma$ in both models in Theorem 1 is possible in closed form, since both Wishart and Inverse Wishart distributions are conjugate priors for the normal distribution, as was done by Shah et al. (2014) for (ii). However, allowing for arbitrary kernel functions for the output neuron makes this integration no longer tractable. As a workaround, we *learn* the latent kernel $\mathbf{K_{BB}}$ from data, instead of integrating it out. We approximate the GPs in our network by sparse GPs using the FITC approximation (Snelson and Ghahramani, 2006), and infer the model posterior using variational inference. The resultant sparse approximation to DFGP reads

$$
\begin{aligned}
p(\mathbf{A}|\mathbf{X}) &= \prod_{r=1}^{\nu} \mathcal{N}(\mathbf{a}_r|\mathbf{0}, \mathbf{K}_{\mathbf{X}_{ir}\mathbf{X}_{ir}}), \\
p(\mathbf{B}|\mathbf{A}) &= \prod_{n}^{N}\prod_{r=1}^{\nu} \mathcal{N}(b_{nr}|\mathbf{k}_{\mathbf{X}_{ir}\mathbf{x}_n}^T \mathbf{K}_{\mathbf{X}_{ir}\mathbf{X}_{ir}}^{-1}\mathbf{a}_r, k_{\mathbf{x}_n\mathbf{x}_n} - \mathbf{k}_{\mathbf{X}_{ir}\mathbf{x}_n}^T \mathbf{K}_{\mathbf{X}_{ir}\mathbf{X}_{ir}}^{-1}\mathbf{k}_{\mathbf{X}_{ir}\mathbf{x}_n}), \\
p(\mathbf{u}|\mathbf{Z}) &= \mathcal{N}(\mathbf{u}|\mathbf{0}, \mathbf{K_{ZZ}}), \\
p(\mathbf{f}|\mathbf{B}, \mathbf{Z}, \mathbf{u}) &= \prod_{n}^{N} \mathcal{N}(f_n|\mathbf{k}_{\mathbf{Z}\mathbf{b}_n}^T \mathbf{K_{ZZ}}^{-1}\mathbf{u}, k_{\mathbf{b}_n\mathbf{b}_n} - \mathbf{k}_{\mathbf{Z}\mathbf{b}_n}^T \mathbf{K_{ZZ}}^{-1}\mathbf{k}_{\mathbf{Z}\mathbf{b}_n}), \\
p(\mathbf{y}|\mathbf{f}) &= \mathcal{N}(\mathbf{y}|\mathbf{f}, \beta^{-1}\mathbf{I}),
\end{aligned}
$$

where $\mathbf{Z} = [\mathbf{z}_1; \cdots; \mathbf{z}_P]$ has the inducing points in its rows, and $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_\nu]$ has the latent inducing vectors of the first-level GPs in its columns. The matrix $\mathbf{X}_{ir}$ is a subset of $\mathbf{X}$ used as inducing points for DoF $r$, and $\mathbf{B} = [\mathbf{b}_1; \cdots; \mathbf{b}_N]$ is the latent output of the first-level GPs. The vector $\mathbf{u}$ is the latent inducing vector of the second-level GP that merges the first-level GPs through the latent inducing covariance $\mathbf{K_{ZZ}}$, the vector $\mathbf{f}$ is the latent noiseless output of the second-level GP, and $\mathbf{y}$ is the observed output with additive noise.

We approximate the posterior by the factorized distribution

$$
Q = p(\mathbf{f}|\mathbf{B}, \mathbf{Z}, \mathbf{u}) q(\mathbf{u}) \prod_{n}^{N} p(\mathbf{b}_n|\mathbf{A}) \prod_{r=1}^{\nu} q(\mathbf{a}_r)
$$

with $q(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{m}, \mathbf{S})$, and $q(\mathbf{a}_r) = \mathcal{N}(\mathbf{a}_r|\mathbf{c}_r, \mathbf{D}_r)$. This type of a factorization for sparse GPs (SGPs) has been proposed earlier by Snelson and Ghahramani (2006). Later on,

Titsias and Lawrence (2010) and Damianou and Lawrence (2013) used extensions of it for Bayesian GPLVM and Deep GP, respectively. Using Jensen's inequality, the log-marginal distribution can be lower bounded as

$$\log\, p(\mathbf{y}|\mathbf{Z},\mathbf{X}) \geq \mathcal{L}_r = \mathbb{E}_Q\left[\log \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{B},\mathbf{Z},\mathbf{u})p(\mathbf{u}|\mathbf{Z})p(\mathbf{B}|\mathbf{A})p(\mathbf{A}|\mathbf{X})}{Q}\right]$$
$$= \mathbb{E}_Q[\log p(\mathbf{y}|\mathbf{f})] - \mathbb{KL}\Big(q(\mathbf{u})\Big\|p(\mathbf{u}|\mathbf{Z})\Big) - \sum_{r=1}^{\nu}\mathbb{KL}\Big(q(\mathbf{a}_r)\Big\|p(\mathbf{a}_r|\mathbf{X})\Big),$$

where $\mathcal{L}_r$ is the variational lower bound and $\mathbb{KL}(\cdot||\cdot)$ is the Kullback-Leibler divergence between the two distributions in the arguments. The essence of this formulation is that the term $p(\mathbf{f}|\mathbf{B},\mathbf{Z},\mathbf{u})$ that makes the number of model parameters grow with the training set size is cancelled out by keeping it identical in the approximate distribution $Q$. Note here that the inducing points $\mathbf{Z}$ are assumed given, hence are treated as variational parameters and learned from data, as suggested by Titsias and Lawrence (2010). The resultant variational lower bound reads

$$\mathcal{L}_r = \beta\mathbf{y}^T\mathbb{E}_{Q_{AB}}[\mathbf{K_{ZB}}]^T\mathbf{K_{ZZ}^{-1}}\mathbf{m} - \frac{\beta}{2}tr\{\mathbf{K_{ZZ}^{-1}}\mathbb{E}_{Q_{AB}}[\mathbf{K_{ZB}}\mathbf{K_{ZB}^T}]\mathbf{K_{ZZ}^{-1}}(\mathbf{m}\mathbf{m}^T+\mathbf{S})\} + \frac{1}{2}\log|\mathbf{S}|$$
$$- \frac{\beta}{2}tr\{\mathbb{E}_{Q_{AB}}[\mathbf{K_{BB}}]\} + \frac{\beta}{2}tr\{\mathbf{K_{ZZ}^{-1}}\mathbb{E}_{Q_{AB}}[\mathbf{K_{ZB}}\mathbf{K_{ZB}^T}]\} - \frac{1}{2}\log|\mathbf{K_{ZZ}}| - \frac{1}{2}\mathbf{m}^T\mathbf{K_{ZZ}^{-1}}\mathbf{m}$$
$$- \frac{1}{2}tr(\mathbf{K_{ZZ}^{-1}}\mathbf{S}) - \frac{1}{2}\sum_{r=1}^{\nu}tr\{\mathbf{K_{\mathbf{X}_{ir}\mathbf{X}_{ir}}^{-1}}\mathbb{E}_{q(\mathbf{a}_r)}[\mathbf{a}_r\mathbf{a}_r^T]\} + \frac{1}{2}\sum_{r=1}^{\nu}\log|\mathbf{D}_r| + \frac{N}{2}\log\beta - \frac{\beta}{2}\mathbf{y}^T\mathbf{y},$$

where $Q_{AB} = p(\mathbf{B}|\mathbf{A})q(\mathbf{A})$. Here, we need to calculate $\mathbb{E}_{p(\mathbf{B}|\mathbf{A})}[\mathbf{K_{ZB}}]$, $\mathbb{E}_{p(\mathbf{B}|\mathbf{A})}[\mathbf{K_{\mathbf{b}_n\mathbf{b}_n}}]$, and $\mathbb{E}_{p(\mathbf{B}|\mathbf{A})}[\mathbf{K_{ZB}}\mathbf{K_{ZB}^T}]$. For Gaussian kernel functions $k(\mathbf{z},\mathbf{x}) = \exp\{-\frac{1}{2}(\mathbf{z}-\mathbf{x})^T\mathbf{J}^{-1}(\mathbf{z}-\mathbf{x})\}$, such as RBF ($\mathbf{J}=\gamma\mathbf{I}$), all these three integrals are available in closed form (see appendix). Taking the expectation of these kernel entries with respect to a normally-distributed $q(\mathbf{A})$ is also a Gaussian integral, hence tractable. However, in order not to increase the number of matrix multiplications, we approximate $q(\mathbf{a}_r)$ by an infinitesimal spike at its mean $\mathbf{c}_r$. This way, we can take the expectation of $q(\mathbf{A})$ simply by replacing each $b_{nr}$ with $\mathbf{k}^T_{\mathbf{X}_{ir}\mathbf{x}_n}\mathbf{K}^{-1}_{\mathbf{X}_{ir}\mathbf{X}_{ir}}\mathbf{c}_r$. Note that differently from the Deep GP, we do not introduce white noise on the latent representation $\mathbf{B}$. Damianou and Lawrence (2013) use this additional latent variable as an independent factor in the variational distribution. This makes inference calculation easier, but introduces $N\times(\nu+\nu^2)$ more variational parameters! By skipping this factor and approximating $q(\mathbf{a}_r)$ by a point estimate, we fix the number of parameters to $P\times\nu$ regardless of the data size. Note also that DFGP remains tractable even when the RBF kernel with full covariance $\mathbf{J}$ is used. This allows metric learning on the latent manifold space. For categorical output, we squeeze the real-valued $\mathbf{y}$ using a Bernoulli-Probit likelihood and marginalize out $\mathbf{y}$. The marginal likelihood can then be bounded by $p(\mathbf{t}|\mathbf{Z},\mathbf{X}) \geq \int \exp(\mathcal{L}_r)p(\mathbf{t}|\mathbf{y})d\mathbf{y}$, as suggested by Hensman et al. (2013). Further details on the update rules can be found in the appendix.

## 5. Experiments

We evaluated DFGP on three different applications. In the first two, we compared it to: i) **SGP:** Sparse GP (Snelson and Ghahramani, 2006) to show the effect of cascading multiple

GPs, hence, learning a latent manifold, ii) **DFGP-WP:** DFGP with a linear kernel on the output GP, equivalent to a WP prior on its covariance matrix, and iii) **Student-t:** The Student-t process (Shah et al., 2014) with the same DoF as the number of hidden neurons we used for DFGP.

We used 10 inducing points for all sparse GPs of all models. For DGFP, we set the inducing points of the GPs on the hidden layer to randomly chosen subsets of the training set in order to cause diversity in the latent representation of the instances. We initialized $\mathbf{c}_r$ and $\mathbf{m}$ and the inducing output mean of SGP by $\hat{\mathbf{m}} = \underset{\mathbf{m}}{\text{argmin}} \ ||\mathbf{K}_{\mathbf{ZX}}^T \mathbf{K}_{\mathbf{ZZ}}^{-1} \mathbf{m} - \mathbf{f}||_2^2$, where $\mathbf{m}$ is the inducing output mean of any sparse GP, $\mathbf{K}_{\mathbf{ZZ}}$ the Gram matrix of the inducing points, $\mathbf{K}_{\mathbf{ZX}}$ the Gram matrix between the inducing points and the training set, and $\mathbf{f}$ the target output. This formulation, essentially, is the least-squares fit of the GP predictive mean to the output. We initialized the inducing points of SGP to cluster centroids learned by k-means, as in Hensman et al. (2013). We applied the same procedure on the initial state of $\mathbf{B}$ for the DFGP output layer inducing points. For both models, these inducing points are updated by gradient ascent during training. We trained all models by 100 iterations.

### 5.1 Urban $SO_2$ concentration level prediction

The task in this public data set [2] is to predict the urban $SO_2$ concentration level 24 hours in advance, given 27 features including the current $SO_2$ concentration and some meteorological measurements. We have chosen 2000 data points randomly from this data set and used equal-sized train/test splits. We repeated this subsampling procedure 20 times and reported the Normalized Mean Squared Error (NMSE) for our model and the baselines in Table 1. DFGP gives prediction error lower than all three baselines with statistical significance (paired t-test, $p < 0.05$). DFGP-WP and Student-t suffer from the low dimensional feature map used by the output neuron and perform clearly worse than DFGP.

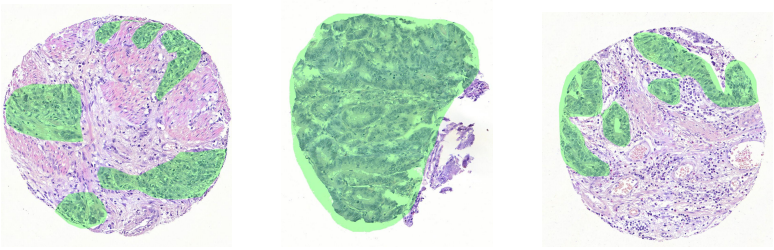
Table 1: **$SO_2$ Data Set Results:** Normalized mean squared errors (NMSE) of the models in comparison.

| | **DFGP** | **DFGP-WP** | **SGP** | **Student-t** |
|---|---|---|---|---|
| **NMSE** | $\mathbf{0.59 \pm 0.03}$ | $0.68 \pm 0.04$ | $0.61 \pm 0.03$ | $0.67 \pm 0.04$ |

### 5.2 Barrett's cancer prognosis from histology images

This data set contains 178 Tissue Micro Array (TMA) images taken from the biopsy samples of 77 Barrett's cancer patients. Tumor regions of the sample images are marked by expert pathologists. Automated detection of tumor regions for this same cohort has previously been studied by Kandemir et al. (2014). Here we study the harder task of prognosing cancer from the same TMA images. More specifically, we predict the patient's survival time (how many months the patient will live after the biopsy sample is taken). Being an objective measure of cancer severity, as opposed to the subjective grading schemes such as Gleason grading (Gleason, 1992), survival time prediction provides crucial information to pathologists about the stage of the cancer, which is critical for the choice of the treatment method. The main

2. `http://theoval.cmp.uea.ac.uk/~gcc/competition/`



**Poor survival** 1.38 months | **Medium survival** 75.22 months | **Good survival** 135.75 months

Figure 2: Biopsy samples of three Barrett's cancer patients with different survival times.

challenge in this task is that it is hard to define visual indicators for survival. Consider the example in Figure 2 where three TMA samples from three patients with different survival profiles are provided with tumor regions shown in green. A tissue surrounded entirely by tumor (middle) could come from a patient with better survival than one with less tumor regions (left). As another example, a tissue with glandular structures (circular groups of cells), which are indicators of a higher level of Barrett's cancer progression (right), could come from a patient with much better survival than a patient without glandular cancer cells (left).

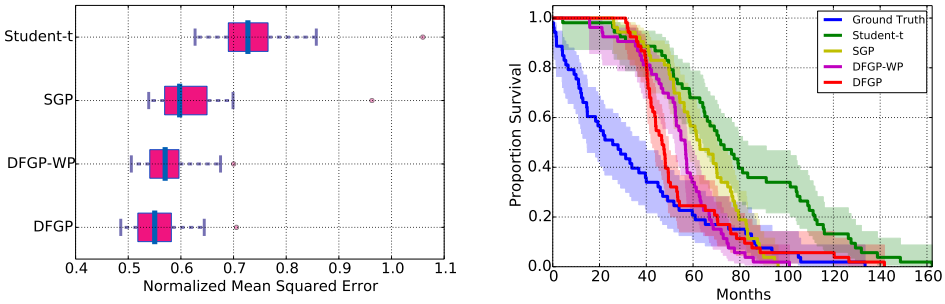



Figure 3: Barrett's cancer survival time prediction. NMSE on the left evaluates patient-level prediction performance, while the Kaplan-Meier curve on the right shows the success of the models in predicting the survival characteristics of the disease at the cohort level.

We split the tumor region of each TMA image into a regular grid of 200×200 pixels and represent each patch by a 738-dimensional feature vector consisting of the intensity, texture, and cell morphology features described in Kandemir et al. (2014). The resultant data set consists of 6573 instances. We reduce the data dimensionality to 50 using principal component analysis (PCA) both to eliminate uninformative features and to prevent overparameterization of SGP, which has one parameter for each dimension of each inducing input point. To simulate the extreme scarcity of real-world survival data, we use only a randomly chosen 25% of the data set for training and the rest for evaluation. The data set is split at the patient level (i.e. all instances belonging to a patient are put either into the training or the evaluation split). We repeat this procedure 20 times. Since a patient is represented

with multiple instances, we predicted her survival time by averaging the predictions made on all patches of her samples. We set the number of hidden-layer GPs of DFGP to 1/4 th of the input dimensionality.

Figure 3 shows the NMSE (left) and the Kaplan-Meier (KM) curves (right) of DFGP and the baselines. DFGP gives lower NMSE than all the other models in patient-level survival prediction with statistical significance (paired t-test, $p < 0.05$). The KM curve, on the other hand, shows the survival characteristics of the disease on the entire cohort. Hence, a better KM curve (one that is more similar to the ground-truth curve) does not necessarily imply better patient-level prediction. Similarity of the curves of all models to the ground truth in the figure indicate that the survival characteristic of the disease on a patient population could be predicted to a reasonable extent from biopsy images without requiring several years of patient monitoring. The curve for DFGP is more similar to the ground-truth than the baselines. DFGP runs on a training split of this data set ($\approx$ 1800 data points, depending on the split) in 161 seconds and predicts on a test split ($\approx$ 4800 data points) in 0.06 seconds.

### 5.3 Cell event detection by multiple kernel learning

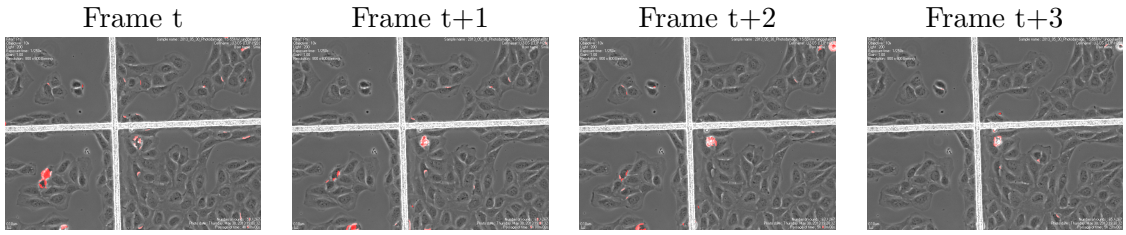



Figure 4: Detected event candidate regions from an example time-lapse phase-contrast image sequence of a human osteosarcoma cell culture. There is an apoptosis in the middle and a mitosis at the top right corner.

It is trivial to adapt DFGP to the MKL setting. As a proof-of-concept study, we detect mitosis (division) and apoptosis (death) of live human osteosarcoma cells from time-lapse phase-contrast microscopy image sequences. The data includes 8 sequences with 134 images each. We detected candidate regions from each frame to contain events by taking the pixel-wise intensity difference of consecutive frames, thresholding by 1/4 th of the maximum intensity difference, and removing the connected components smaller than 20 pixels. Figure 4 shows four consecutive frames from an example sequence where candidate regions are marked as red. The region corresponding to the apoptosis event at the image center is very small before the event (Frame t), grows when the event starts (Frames t+1 and t+2), and shrinks back immediately (Frame 4). The mitosis at the top right corner also follows the same pattern. Each candidate region is represented by features extracted from the 52×52-pixel patch surrounding its centroid in the previous frame and the current frames. An event candidate is represented by a 26-bin intensity histogram calculated on the intensity difference of that frame and its predecessor, 58 Local Binary Pattern (LBP) features of the current frame, and another 58 LBP features and 128 Scale Invariant Feature Transform (SIFT) features for the difference of it from the predecessor frame. We treat each of the

four feature sets and the two time points as separate views, and assign each of these eight data views a RBF kernel. We consider mitosis and apoptosis events as one single class and the rest as another class. For DFGP, we assign each view to one hidden neuron (i.e. $\nu = 8$).

We compare DFGP to the following three baselines: i) **BEMKL**: a Bayesian MKL model (Gönen, 2012) that infers a linear combination of kernels from data, ii) **MKGP**: an alternative GP-based MKL model (Lázaro-Gredilla and Titsias, 2011) that assigns a GP to each view and linearly combines them into a single output using a spike-and-slab prior on the kernel weights, and iii) **SVM-UA**: Support Vector Machine (SVM) with uniform average (UA) of kernels $\frac{1}{\nu}\sum_{r=1}^{\nu}\mathbf{K}_r$, meant for illustrating the effect of MKL.

We considered each sequence as a data split and replicated our experiments for each possible training-evaluation sequence pair (i.e. 56 times). We report the average F1 Scores and standard deviations across all 56 splits in Table 2. The fact that DFGP improves on SVM-UA serves as a proof-of-concept for its usability for MKL. Its being better than BEMKL and MKGP is due to its additional flexibility coming from the non-linear combination of kernels in Equation 5.

Table 2: Cell event (mitosis and apoptosis) detection results.

| | **DFGP** | **BEMKL** | **MKGP** | **SVM-UA** |
|---|---|---|---|---|
| Avg F1 Score | $\mathbf{0.80 \pm 0.09}$ | $0.79 \pm 0.11$ | $0.70 \pm 0.18$ | $0.77 \pm 0.12$ |

## 6. Discussion

We attribute the performance improvement of DFGP over SGP to the additional expressive power coming from the latent manifold. Although we restricted our analysis on two layers in order to make sense of the resultant model in terms of covariance priors, extending it to more layers is trivial, and benchmarking its effectiveness is left to future work. Referring to earlier studies that report saturation in performance after few layers when kernelized neurons are used (Mairal et al., 2014), we expect a shallow DFGP to perform comparable both to its own deeper architectures and conventional deep neural nets. Observing this outcome would arise the following fundamental question about basic principles of deep learning: *"Should we build deep architectures of simple neurons or shallow architectures of complex neurons?"*. As opposed to the simple learning scheme (backpropagation) of the former, which enjoys easy and robust implementations, the novel latter would have much less number of model parameters to be learned using more complicated algorithms.

In MKL (Gönen and Alpaydın, 2011), the mainstream approach is to combine kernels linearly. Even though non-linear combinations in simple forms, such as the weighted Hadamard product of kernels, are proposed, the level of non-linearity provided by the RBF kernel has not been previously reached. The effect of this non-linearity on performance could be benchmarked and quantified in a wider spectrum of applications, such as in cases where each kernel corresponds to a different data modality (e.g. image and gene sequence), or when there are hundreds of kernels to be combined.

## References

A.C. Damianou and N.D. Lawrence. Deep Gaussian processes. In *AISTATS*, 2013.

A.P. Dawid. Some matrix-variate distribution theory: Notational considerations and a Bayesian application. *Biometrika*, 68(1):265–274, 1981.

D.F. Gleason. Histologic grading of prostate cancer: A perspective. *Human pathology*, 23 (3):273–279, 1992.

M. Gönen. Bayesian efficient multiple kernel learning. In *ICML*, 2012.

M. Gönen and E. Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12(Jul):2211–2268, 2011.

J. Hensman, N. Fusi, and N.D. Lawrence. Gaussian processes for big data. In *UAI*, 2013.

N. Houlsby, F. Huszar, Z. Ghahramani, and J.M. Hernández-Lobato. Collaborative Gaussian processes for preference learning. In *NIPS*, 2012.

M. Kandemir, A. Feuchtinger, A. Walch, and F. A. Hamprecht. Digital Pathology: Multiple instance learning can detect Barrett's cancer. In *ISBI*, 2014.

M. Kandemir, C. Wojek, and F. A. Hamprecht. Cell Event Detection in Phase-Contrast Microscopy Sequences from Few Annotations . In *MICCAI*, 2015.

M. Lázaro-Gredilla and M.K. Titsias. Spike and slab variational inference for multi-task and multiple kernel learning. In *NIPS*, 2011.

W-J. Li, Z. Zhang, and D.-Y. Yeung. Latent Wishart processes for relational kernel learning. In *AISTATS*, 2009.

J. Mairal, P. Koniusz, Z. Harchaoui, and C. Schmid. Convolutional kernel networks. In *NIPS*, 2014.

R.A. Pacheco, J. Hensman, M. Zwiessele, and N.D. Lawrence. Hybrid discriminative-generative approach with Gaussian processes. In *AISTATS*, 2014.

C.E. Rasmussen and C.I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

M. Seeger. Bayesian Gaussian process models: PAC-Bayesian generalisation error bounds and sparse approximations. *PhD Thesis*, 2003.

A. Shah, A.G. Wilson, and Z. Ghahramani. Student-t processes as alternatives to Gaussian processes. In *AISTATS*, 2014.

A.J. Smola and B. Schölkopf. *Learning with kernels*. MIT Press, 1998.

E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *NIPS*, 2006.

E. Snelson, C.E. Rasmussen, and Z. Ghahramani. Warped Gaussian processes. In *NIPS*, 2004.

M.K. Titsias and N.D. Lawrence. Bayesian Gaussian process latent variable model. In *AISTATS*, 2010.

A.G. Wilson and Z. Ghahramani. Generalised Wishart processes. In *UAI*, 2011.

Z. Zhang, J.T. Kwok, and D.Y. Yeung. Model-based transductive learning of the kernel matrix. In *Machine Learning*, pages 69–101, 2006.

## Appendix

### Variational Lower Bound for Classification

Given the variational lower bound $\mathcal{L}_r$ for continuous output, the lower bound for binary output can be calculated by adding the Bernoulli-Probit likelihood

$$p(\mathbf{t}|\mathbf{y}) = \prod_{n=1}^{N} Bernoulli(t_n|\Phi(y_n))$$

to the model, and marginalizing out $\mathbf{y}$. The marginal likelihood for the GP classifier can be bounded by $p(\mathbf{t}|\mathbf{Z},\mathbf{X}) \geq \int \exp(\mathcal{L}_r)p(\mathbf{t}|\mathbf{y})d\mathbf{y}$. After taking this integral, the lower bound becomes

$$\begin{aligned}\log p(\mathbf{t}|\mathbf{Z},\mathbf{X}) \geq \mathcal{L}_c = \mathcal{L}_r + \sum_{n}^{N} t_n \log \Phi\Big(\frac{\mathbf{m}^T\mathbf{K}_{\mathbf{ZZ}}^{-1}\mathbb{E}[\mathbf{K}_{\mathbf{Zb}_n}]}{\sqrt{\beta^{-1}+1}}\Big) \\ + \sum_{n}^{N} t_n \mathbb{I}(t_n=1)\frac{\beta}{2}(\mathbf{m}^T\mathbf{K}_{\mathbf{ZZ}}^{-1}\mathbb{E}[\mathbf{K}_{\mathbf{ZB}}])^2 + \sum_{n}^{N} t_n \mathbb{I}(t_n=-1)\log\sqrt{\frac{2\pi}{\beta}},\end{aligned}$$

where $\mathbb{I}(\cdot)$ is the indicator function.

### Variational Update Rules

For regression, a mean-field update is tractable for $q(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{m},\mathbf{S})$ as follows

$$\begin{aligned}\mathbf{S} &= \Big(\mathbf{K}_{\mathbf{ZZ}}^{-1} + \beta\sum_{n}^{N} tr\{\mathbf{K}_{\mathbf{ZZ}}^{-1}\mathbb{E}_{Q_{AB}}[\mathbf{k}_{\mathbf{ZB}}\mathbf{k}_{\mathbf{ZB}}^T]\mathbf{K}_{\mathbf{ZZ}}^{-1}\}\Big)^{-1}, \\ \mathbf{m} &= \beta\mathbf{S}\mathbf{K}_{\mathbf{ZZ}}^{-1}\mathbb{E}_{Q_{AB}}[\mathbf{k}_{\mathbf{ZB}}]\mathbf{y}.\end{aligned}$$

However, for classification, this update should be done gradient-based, since $\mathbf{m}$ also appears in the Bernoulli-Probit likelihood in a non-conjugate way. The related gradient equations

are

$$\begin{aligned}\frac{\partial \mathcal{L}_c}{\partial \mathbf{m}} =& -\beta \mathbf{K_{ZZ}}^{-1}\mathbb{E}_{Q_{AB}}[\mathbf{K_{ZB}}\mathbf{K_{ZB}}^T]\mathbf{K_{ZZ}}^{-1}\mathbf{m} - \mathbf{K_{ZZ}}^{-1}\mathbf{m} \\ &+ \sum_{n=1}^{N} \mathbb{I}(t_n = 1)\beta\Big(\mathbf{m}^T\mathbf{K_{ZZ}}^{-1}\mathbb{E}_{Q_{AB}}[\mathbf{K_{Zb}}_n]\Big)\mathbf{K_{ZZ}}^{-1}\mathbb{E}_{Q_{AB}}[\mathbf{K_{Zb}}_n] \\ &+ \sum_{n=1}^{N} t_n \frac{\mathbf{K_{ZZ}}^{-1}\mathbb{E}_{Q_{AB}}[\mathbf{K_{Zb}}_n]}{\sqrt{2\pi}(\beta^{-1}+1)\Phi\left(\frac{\mathbf{m}^T\mathbf{K_{ZZ}}^{-1}\mathbb{E}_{Q_{AB}}[\mathbf{K_{Zb}}_n]}{\sqrt{\beta^{-1}+1}}\right)},\end{aligned}$$

and

$$\frac{\partial \mathcal{L}_c}{\partial \mathbf{S}} = -\frac{1}{2}\mathbf{K_{ZZ}}^{-1} + \frac{1}{2}\mathbf{S}^{-T} - \frac{\beta}{2}\mathbf{K_{ZZ}}^{-1}\mathbb{E}_{Q_{AB}}[\mathbf{K_{ZB}}\mathbf{K_{ZB}}^T]\mathbf{K_{ZZ}}^{-1}.$$

For both regression and classification, the gradient of the lower bound with respect to $\mathbf{c}_r$ is

$$\begin{aligned}\frac{\partial \mathcal{L}_r}{\partial \mathbf{c}_r} =& -\mathbf{K}^{-1}_{\mathbf{X}_{ir}\mathbf{X}_{ir}}\mathbf{c}_r + \beta\mathbf{y}^T\frac{\partial \mathbb{E}_{Q_{AB}}[\mathbf{K_{ZB}}]^T}{\partial \mathbf{c}_r}\mathbf{K_{ZZ}}^{-1}\mathbf{m} \\ &- \frac{\beta}{2}tr\Big\{\mathbf{K_{ZZ}}^{-1}\frac{\partial \mathbb{E}_{Q_{AB}}[\mathbf{K_{ZB}}\mathbf{K_{ZB}}^T]}{\partial \mathbf{c}_r}\mathbf{K_{ZZ}}^{-1}(\mathbf{m}\mathbf{m}^T + \mathbf{S})\Big\} \\ &- \frac{\beta}{2}tr\Big\{\frac{\partial \mathbb{E}_{Q_{AB}}[\mathbf{K_{BB}}]}{\partial \mathbf{c}_r} - \mathbf{K_{ZZ}}^{-1}\frac{\partial \mathbb{E}_{Q_{AB}}[\mathbf{K_{ZB}}\mathbf{K_{ZB}}^T]}{\partial \mathbf{c}_r}\Big\}.\end{aligned}$$

We learn the inducing points by optimizing the lower bound with respect to each entry of $\mathbf{Z}$. The derivative of the lower bound with respect to the inducing point $p$ of DoF $r$ for regression is

$$\begin{aligned}\frac{\partial \mathcal{L}_r}{\partial z_{pr}} =& \beta\mathbf{y}^T\Big(\frac{\partial \mathbb{E}_{Q_{AB}}[\mathbf{K_{ZB}}_n]}{\partial z_{pr}}^T\mathbf{K_{ZZ}}^{-1} + \mathbb{E}_{Q_{AB}}[\mathbf{K_{ZB}}_n]^T\frac{\mathbf{K_{ZZ}}^{-1}}{\partial z_{pr}}\Big)\mathbf{m} \\ &- \frac{\beta}{2}tr\Bigg\{\Big(\frac{\partial \mathbf{K_{ZZ}}^{-1}}{\partial z_{pr}}\mathbb{E}_{Q_{AB}}[\mathbf{K_{ZB}}\mathbf{K_{ZB}}^T]\mathbf{K_{ZZ}}^{-1} + \mathbf{K_{ZZ}}^{-1}\frac{\partial \mathbb{E}_{Q_{AB}}[\mathbf{K_{ZB}}\mathbf{K_{ZB}}^T]}{\partial z_{pr}}\mathbf{K_{ZZ}}^{-1} \\ &\qquad + \mathbf{K_{ZZ}}^{-1}\mathbb{E}_{Q_{AB}}[\mathbf{K_{Zb}}_n\mathbf{K_{Zb}}_n^T]\frac{\partial \mathbf{K_{ZZ}}^{-1}}{\partial z_{pr}}\Big)(\mathbf{m}\mathbf{m}^T + \mathbf{S})\Bigg\} \\ &+ \frac{\beta}{2}tr\Bigg\{\frac{\mathbf{K_{ZZ}}^{-1}}{\partial z_{pr}}\mathbb{E}_{Q_{AB}}[\mathbf{K_{ZB}}\mathbf{K_{ZB}}^T] + \mathbf{K_{ZZ}}^{-1}\frac{\partial \mathbb{E}_{Q_{AB}}[\mathbf{K_{ZB}}\mathbf{K_{ZB}}^T]}{\partial z_{pr}}\Bigg\} \\ &- \frac{1}{2}tr\Big(\mathbf{K_{ZZ}}^{-1}\frac{\partial \mathbf{K_{ZZ}}}{\partial z_{pr}} + \frac{\partial \mathbf{K_{ZZ}}^{-1}}{\partial z_{pr}}\mathbf{S}\Big) - \frac{1}{2}\mathbf{m}^T\frac{\partial \mathbf{K_{ZZ}}^{-1}}{\partial z_{pr}}\mathbf{m}.\end{aligned} \tag{7}$$

For classification, this derivative is

$$\frac{\partial \mathcal{L}_c}{\partial z_{pr}} = \frac{\partial \mathcal{L}_r}{\partial z_{pr}} + \sum_{n=1}^{N} t_n\frac{\mathcal{N}(F_n|0,1)}{\Phi(F_n)} + \sum_{n=1}^{N} t_n\frac{\partial F_n}{\partial z_{pr}} + \sum_{n=1}^{N}\mathbb{I}(t_n = 1)\beta F_n\frac{\partial F_n}{\partial z_{pr}}, \tag{8}$$

where $F_n = \frac{\mathbf{m}^T \mathbf{K}_{\mathbf{ZZ}}^{-1} \mathbb{E}_{Q_{AB}}[\mathbf{K}_{\mathbf{Zb}_n}]}{\sqrt{\beta^{-1}+1}}$ and

$$\frac{\partial F_n}{\partial z_{pr}} = \mathbf{m}^T \left( \frac{\partial \mathbf{K}_{\mathbf{ZZ}}^{-1}}{\partial z_{pr}} \mathbb{E}_{Q_{AB}}[\mathbf{K}_{\mathbf{Zb}_n}] + \mathbf{K}_{\mathbf{ZZ}}^{-1} \frac{\partial \mathbb{E}_{Q_{AB}}[\mathbf{K}_{\mathbf{Zb}_n}]}{\partial z_{pr}} \right).$$

Given a Gaussian kernel function $k(\mathbf{z}, \boldsymbol{\mu}) = \exp\{-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^T \mathbf{J}^{-1}(\mathbf{z}-\boldsymbol{\mu})\}$, which could be isotropic as in the radial basis function (RBF), diagonal as in Automatic Relevance Determination (ARD), or a full matrix as in metric learning, hyperparameters $\mathbf{J}$ can also be fit by gradient ascent. The derivatives of the variational log-likelihood with respect to the kernel hyperparameters are as in Equations 7 and 8. It suffices to replace all $\partial z_{pr}$s in these formulas with $\partial \mathbf{J}_{ij}$.

### RBF Kernel for Random Inputs

For a Radial Basis Function $k(\mathbf{x}, \mathbf{x}') = \exp\{-\frac{1}{2}(\mathbf{x}-\mathbf{x}')^T \mathbf{J}^{-1}(\mathbf{x}-\mathbf{x}')\}$, static input vectors $\mathbf{z}$ and $\mathbf{z}'$, and the random vector $\mathbf{x}$ which follows the multivariate normal distribution $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$,

$$\mathbb{E}_{p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})}[k(\mathbf{z}, \mathbf{x})] = |\mathbf{J}^{-1}\boldsymbol{\Sigma} + \mathbf{I}|^{-\frac{1}{2}} \times \exp\Big\{ -\frac{1}{2}\mathbf{z}^T(\mathbf{J}+\boldsymbol{\Sigma})^{-1}\mathbf{z} - \frac{1}{2}\boldsymbol{\mu}^T(\mathbf{J}+\boldsymbol{\Sigma})^{-1}\boldsymbol{\mu} + \mathbf{z}^T\mathbf{J}^{-1}(\mathbf{J}^{-1}+\boldsymbol{\Sigma}^{-1})^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\Big\},$$

$$\begin{aligned}\mathbb{E}_{p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})}[k(\mathbf{z}_p, \mathbf{x})k(\mathbf{z}_{p'}, \mathbf{x})] = |2\mathbf{J}^{-1}\boldsymbol{\Sigma} + \mathbf{I}|^{-\frac{1}{2}} \times \exp\Big\{ &-\frac{1}{2}\mathbf{z}_p^T(\mathbf{J}^{-1} - \mathbf{J}^{-1}(2\mathbf{J}^{-1}+\boldsymbol{\Sigma}^{-1})^{-1}\mathbf{J}^{-1})\mathbf{z}_p \\ &-\frac{1}{2}\mathbf{z}_{p'}^T(\mathbf{J}^{-1} - \mathbf{J}^{-1}(2\mathbf{J}^{-1}+\boldsymbol{\Sigma}^{-1})^{-1}\mathbf{J}^{-1})\mathbf{z}_{p'} - \frac{1}{2}\boldsymbol{\mu}^T(\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}(2\mathbf{J}^{-1}+\boldsymbol{\Sigma}^{-1})^{-1}\boldsymbol{\Sigma}^{-1})\boldsymbol{\mu} \\ &+ (\mathbf{z}_p + \mathbf{z}_{p'})^T\mathbf{J}^{-1}(2\mathbf{J}^{-1}+\boldsymbol{\Sigma}^{-1})^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \mathbf{z}_p^T\mathbf{J}^{-1}(2\mathbf{J}^{-1}+\boldsymbol{\Sigma}^{-1})^{-1}\mathbf{J}^{-1}\mathbf{z}_{p'}\Big\}.\end{aligned}$$

The derivatives of the stochastic Gaussian kernel with respect to an inducing point entry $z_{pr}$ are

$$\frac{\partial \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})}[k(\mathbf{z}_p, \boldsymbol{\mu})]}{\partial z_{pr}} = \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})}[k(\mathbf{z}_p, \boldsymbol{\mu})]\Big( -\mathbf{z}_p^T\mathbf{J}^{-1} - \mathbf{J}^{-1}(\boldsymbol{\Sigma}^{-1}+\mathbf{J}^{-1})^{-1}\mathbf{J}^{-1} + \boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\Sigma}^{-1}+\mathbf{J}^{-1})^{-1}\mathbf{J}^{-1}\frac{\partial \mathbf{z}_p}{\partial z_{pr}}\Big),$$

and

$$\begin{aligned}\frac{\partial \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})}[k(\mathbf{z}_p, \boldsymbol{\mu})k(\mathbf{z}_{p'}, \boldsymbol{\mu})]}{\partial z_{pr}} = &\mathbb{E}_{p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})}[k(\mathbf{z}_p, \boldsymbol{\mu})k(\mathbf{z}_{p'}, \boldsymbol{\mu})] \\ &\times \Bigg( -\mathbf{z}_p^T(\mathbf{J}^{-1} - \mathbf{J}^{-1}(2\mathbf{J}^{-1}+\boldsymbol{\Sigma}^{-1})^{-1}\mathbf{J}^{-1})\frac{\partial \mathbf{z}_p}{\partial z_{pr}} \\ &+ \Big(\frac{\partial \mathbf{z}_p}{\partial z_{pr}} + \mathbf{z}_{p'}\Big)^T\mathbf{J}^{-1}(2\mathbf{J}^{-1}+\boldsymbol{\Sigma}^{-1})^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \\ &+ \mathbf{z}_{p'}^T\mathbf{J}^{-1}(2\mathbf{J}^{-1}+\boldsymbol{\Sigma}^{-1})^{-1}\mathbf{J}^{-1}\frac{\partial \mathbf{z}_p}{\partial z_{pr}}\Bigg)\end{aligned}$$

for $p \neq p'$. Since $k(\mathbf{z}_p, \mathbf{z}_{p'}) = 1$, this second derivative will be 0 for $p = p'$. For the same reason, the derivatives of $\mathbb{E}_{p(\mathbf{b}_n|\boldsymbol{\mu},\boldsymbol{\Sigma}_n)}[k(\mathbf{b}_n, \mathbf{b}_n)]$ with respect to $z_{pr}$, and $\mathbf{c}_r$ and $\mathbf{J}_{ij}$ are also all 0. The gradients with respect to $\boldsymbol{\mu}$ are

$$\frac{\partial \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}_n)}[k(\mathbf{z}_p, \boldsymbol{\mu})]}{\partial \boldsymbol{\mu}} = \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}_n)}[k(\mathbf{z}_p, \boldsymbol{\mu})]\Big( -(\mathbf{J}+\boldsymbol{\Sigma})^{-1}\boldsymbol{\mu} + \mathbf{z}^T\mathbf{J}^{-1}(\mathbf{J}^{-1}+\boldsymbol{\Sigma}^{-1})^{-1}\boldsymbol{\Sigma}^{-1}\Big).$$

and

$$\begin{aligned}\frac{\partial \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\mu}_n,\boldsymbol{\Sigma}_n)}[k(\mathbf{z}_p, \boldsymbol{\mu})k(\mathbf{z}_{p'}, \boldsymbol{\mu})]}{\partial \boldsymbol{\mu}} =& \mathbb{E}_{p(\boldsymbol{\mu}|\boldsymbol{\mu}_n,\boldsymbol{\Sigma}_n)}[k(\mathbf{z}_p, \boldsymbol{\mu})k(\mathbf{z}_{p'}, \boldsymbol{\mu})] \\ &\times \Big( -(\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}(2\mathbf{J}^{-1}+\boldsymbol{\Sigma}^{-1})^{-1}\boldsymbol{\Sigma}^{-1})\boldsymbol{\mu} \\ &\quad + (\mathbf{z}_p + \mathbf{z}_{p'})^T\mathbf{J}^{-1}(2\mathbf{J}^{-1}+\boldsymbol{\Sigma}^{-1})^{-1}\boldsymbol{\Sigma}^{-1}\Big).\end{aligned}$$

Given the equations above, the derivatives of $\mathbb{E}_{Q_{AB}}[\mathbf{K}_{\mathbf{ZB}}]$ and $\mathbb{E}_{Q_{AB}}[\mathbf{K}_{\mathbf{ZB}}\mathbf{K}_{\mathbf{ZB}}^T]$ with respect to $\mathbf{c}_r$ could simply be taken using $b_{nr} = \mathbf{k}_{\mathbf{X}_{ir}\mathbf{x}_n}^T\mathbf{K}_{\mathbf{X}_{ir}\mathbf{X}_{ir}}^{-1}\mathbf{c}_r$ and

$$\frac{\partial \mathbf{b}_n}{\partial c_{pr}} = \mathbf{e}_p^T\mathbf{K}_{\mathbf{X}_{ir}\mathbf{X}_{ir}}^{-1}\mathbf{c}_r$$

together with the chain rule. Here, $\mathbf{e}_p$ is a $P \times 1$ vector whose $p$th entry is 1 and other entries are 0. The integer $P$ stands for the number of inducing points.

Finally, the gradients with respect to the hyperparameter $\mathbf{J}_{ij}$ are

$$\begin{aligned}\frac{\partial \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})}[k(\mathbf{z}_p, \boldsymbol{\mu})]}{\partial \mathbf{J}_{ij}} =& \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})}[k(\mathbf{z}_p, \boldsymbol{\mu})]\Bigg( -\frac{1}{2}|\mathbf{J}^{-1}\boldsymbol{\Sigma}+\mathbf{I}|^{-\frac{3}{2}}tr\Big((\mathbf{J}^{-1}\boldsymbol{\Sigma}+\mathbf{I})\frac{\partial \mathbf{J}^{-1}}{\partial \mathbf{J}_{ij}}\Big) \\ &- \frac{1}{2}\mathbf{z}^T\frac{\partial \mathbf{J}^{-1}}{\partial \mathbf{J}_{ij}}\mathbf{z} - \mathbf{z}^T\mathbf{J}^{-1}(\mathbf{J}^{-1}+\boldsymbol{\Sigma}^{-1})\frac{\partial \mathbf{J}^{-1}}{\partial \mathbf{J}_{ij}}\mathbf{z} \\ &- \frac{1}{2}\mathbf{z}^T\mathbf{J}^{-1}\frac{\partial \mathbf{J}^{-1}}{\partial \mathbf{J}_{ij}}\mathbf{J}^{-1}\mathbf{z} - \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\frac{\partial \mathbf{J}^{-1}}{\partial \mathbf{J}_{ij}}\boldsymbol{\Sigma}\mathbf{J}^{-1}\boldsymbol{\mu} \\ &+ \mathbf{z}^T\frac{\partial \mathbf{J}^{-1}}{\partial \mathbf{J}_{ij}}(\mathbf{J}^{-1}\boldsymbol{\Sigma}+\mathbf{I})\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \mathbf{z}^T\mathbf{J}^{-1}\frac{\partial \mathbf{J}^{-1}}{\partial \mathbf{J}_{ij}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\Bigg),\end{aligned}$$

and

$$\begin{aligned}\frac{\partial \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})}[k(\mathbf{z}_p, \boldsymbol{\mu})k(\mathbf{z}_{p'}, \boldsymbol{\mu})]}{\partial \mathbf{J}_{ij}} =& \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})}[k(\mathbf{z}_p, \boldsymbol{\mu})k(\mathbf{z}_{p'}, \boldsymbol{\mu})]\times \\ &\Bigg( -\frac{1}{2}|2\mathbf{J}^{-1}\boldsymbol{\Sigma}+\mathbf{I}|^{-\frac{3}{2}}tr\Big((2\mathbf{J}^{-1}\boldsymbol{\Sigma}+\mathbf{I})\frac{\partial \mathbf{J}^{-1}}{\partial \mathbf{J}_{ij}}\Big) - \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\frac{\partial \mathbf{J}^{-1}}{\partial \mathbf{J}_{ij}}\boldsymbol{\Sigma}\mathbf{J}^{-1}\boldsymbol{\mu} \\ &- \frac{1}{2}\mathbf{z}_p^T\Big(\frac{\partial \mathbf{J}^{-1}}{\partial \mathbf{J}_{ij}} + 2\mathbf{J}^{-1}(\mathbf{J}^{-1}+\boldsymbol{\Sigma}^{-1})\frac{\partial \mathbf{J}^{-1}}{\partial \mathbf{J}_{ij}} + \mathbf{J}^{-1}\frac{\partial \mathbf{J}^{-1}}{\partial \mathbf{J}_{ij}}\mathbf{J}^{-1}\Big)\mathbf{z}_p \\ &- \frac{1}{2}\mathbf{z}_{p'}^T\Big(\frac{\partial \mathbf{J}^{-1}}{\partial \mathbf{J}_{ij}} + 2\mathbf{J}^{-1}(\mathbf{J}^{-1}+\boldsymbol{\Sigma}^{-1})\frac{\partial \mathbf{J}^{-1}}{\partial \mathbf{J}_{ij}} + \mathbf{J}^{-1}\frac{\partial \mathbf{J}^{-1}}{\partial \mathbf{J}_{ij}}\mathbf{J}^{-1}\Big)\mathbf{z}_{p'} \\ &+ (\mathbf{z}_p + \mathbf{z}_{p'})^T\Big(\frac{\partial \mathbf{J}^{-1}}{\partial \mathbf{J}_{ij}}(\mathbf{J}^{-1}\boldsymbol{\Sigma}+\mathbf{I})\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \mathbf{J}^{-1}\frac{\partial \mathbf{J}^{-1}}{\partial \mathbf{J}_{ij}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\Big)\Bigg).\end{aligned}$$