Supplementary Material
Random Feature Expansions for Deep Gaussian Processes

Anonymous Authors[1]

A. Additional Experiments

Using the experimental set-up described in Section 4, Figure 1 demonstrates how the competing models perform with regards to the RMSE (or error rate) and MNLL metric when two hidden layers are incorporated into the competing models. The results follow a similar progression to those reported in Figure 3 of the main paper. The DGP-ARC and DGP-RBF models both continue to perform well after introducing this additional layer. However, the results for the regularized DNN are notably inferior, and the degree of overfitting is also much greater. To this end, the MNLL obtained for the MNIST dataset is not shown in the plot as it was vastly inferior to the values obtained using the other methods. DGP-EP was also observed to have low scalability in this regard whereby it was not possible to obtain sensible results for the MNIST dataset using this configuration.

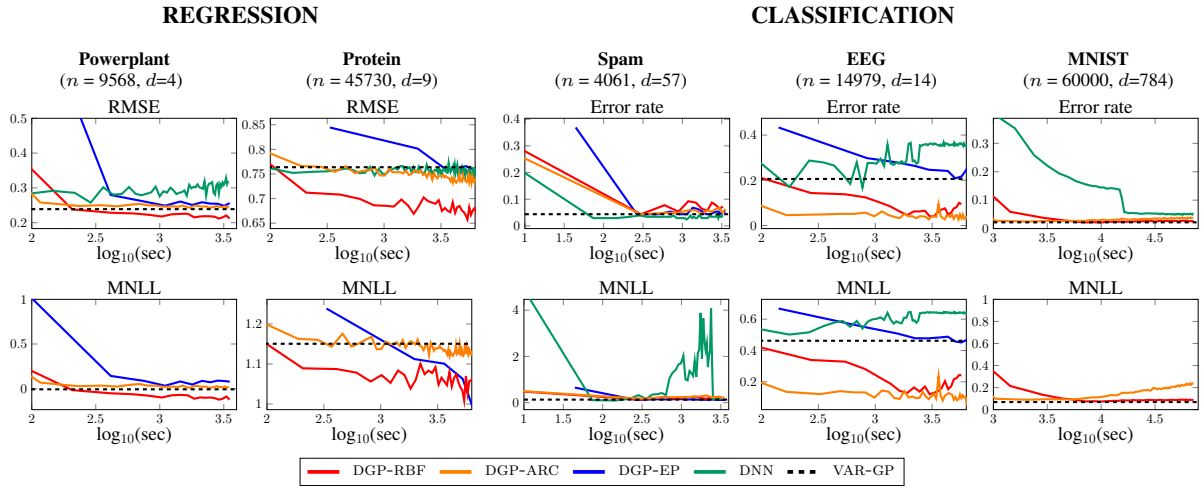

Figure 1. Progression of RMSE and MNLL over time for competing models. Results are shown for configurations having 2 hidden layers. There is no plot for DGP-EP on MNIST because the model did not produce sensible results within the allocated time.

In Section 3.3, we outlined the different strategies for treating $\mathbf{\Omega}$, namely fixing them or treating them variationally, where we observed that the constructed DGP model appears to work best when these are treated variationally while fixing the randomness in their computation throughout the learning process (VAR-FIXED). In Figures 2 and 3, we compare these three approaches on the complete set of datasets reported in the main experiments for one and two hidden layers, respectively. Once again, we confirm that the performance obtained using the VAR-FIXED strategy yields more consistent results than the alternatives. This is especially pertinent to the classification datasets, where the obtained error rate is markedly superior. However, the variation of the model constructed with the ARC-COSINE kernel and optimized using VAR-FIXED appears to be susceptible to some overfitting for higher dimensional datasets (SPAM and MNIST), which is expected given that we are optimizing several covariance parameters (ARD). This would motivate trying to be variational about $\mathbf{\Theta}$ too.

B. Comparison with MCMC

Figure 4 shows a comparison between the variational approximation and MCMC for a two-layer DGP model applied to a regression dataset. The dataset has been generated by drawing 50 data points from $\mathcal{N}(y|h(h(x)), 0.01)$, with $h(x) =$

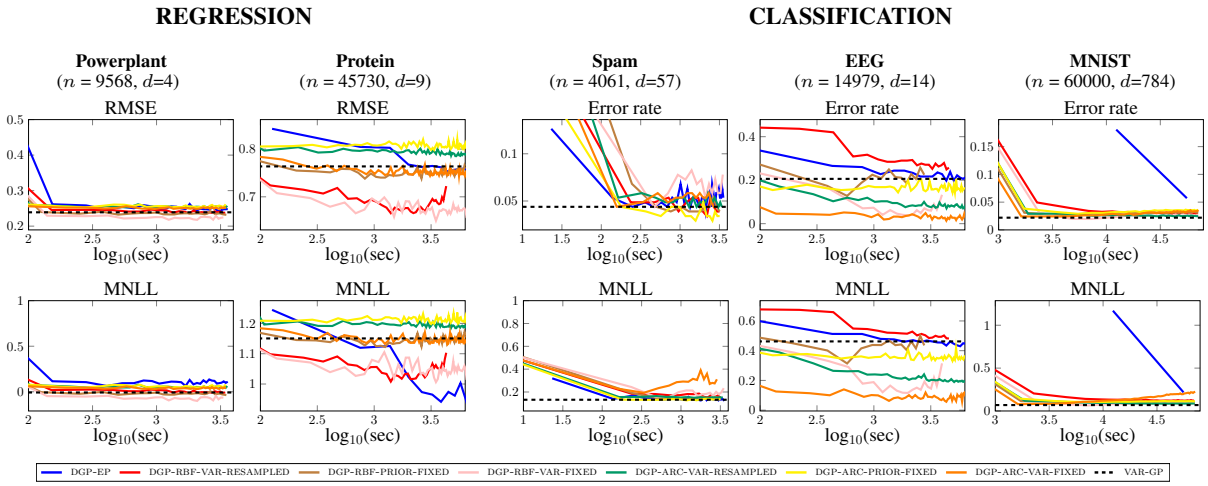

Figure 2. Progression of RMSE and MNLL over time for different optimisation strategies for DGP-ARC and DGP-RBF models. Results are shown for configurations having 1 hidden layer.

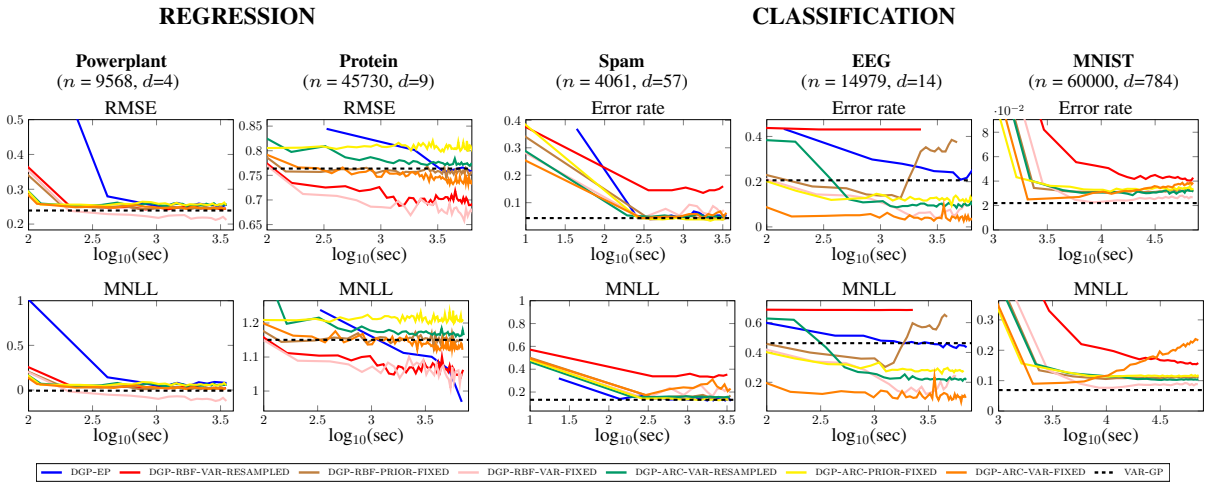

Figure 3. Progression of RMSE and MNLL over time for different optimisation strategies for DGP-ARC and DGP-RBF models. Results are shown for configurations having 2 hidden layers.

$2x\exp(-x^2)$. We experiment with two different levels of precision in the DGP approximation by using 10 and 50 fixed spectral frequencies, respectively, so as to assess the impact on the number of random features on the results. For MCMC, covariance parameters are fixed to the values obtained by optimizing the variational lower bound on the marginal likelihood in the case of 50 spectral frequencies.

We obtained samples from the posterior over the latent variables at each layer using MCMC techniques. In the case of a Gaussian likelihood, it is possible to integrate out the GP at the last layer, thus obtaining a model that only depends on the GP at the first. As a result, the collapsed DGP model becomes a standard GP model whose latent variables can be sampled using various MCMC samplers developed in the literature of MCMC for GPs. Here we employ Elliptical Slice Sampling (Murray et al., *AISTATS*, 2010) to draw samples from the posterior over the latent variables at the first layer, whereas latent variables at the second can be sampled directly from a multivariate Gaussian distribution. More details on the MCMC sampler are reported at the end of this section.

The plots depicted in Figure 4 illustrate how the MCMC approach explores two modes of the posterior of opposite sign. This is due to the output function being invariant to the flipping of the sign of the weights at the two layers. Conversely, the variational approximation over $\mathbf{W}$ accurately identifies one of the two modes of the posterior. The overall approximation is accurate in the case of more random Fourier features, whereas in the case of less, the approximation is unsurprisingly characterized by out-of-sample oscillations. The variational approximation seems to result in larger uncertainty in predictions

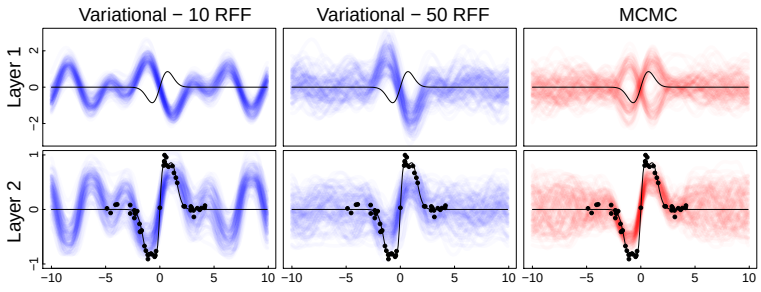

Figure 4. Comparison of MCMC and variational inference of a two-layer DGP with a single GP in the hidden layer and a Gaussian likelihood. The posterior over the latent functions is based on 100 MCMC samples and 100 samples from the variational posterior.

compared to MCMC; we attribute this to the factorization of the posterior over all the weights.

B.1. Details of MCMC sampler for a two-layer DGP with a Gaussian likelihood

We give details of the MCMC sampler that we used to draw samples from the posterior over latent variables in DGPs. In the experiments, we regard this as the gold-standard against which we compare the quality of the proposed DGP approximation and inference. For the sake of tractability, we assume a two-layer DGP with a Gaussian likelihood, and we fix the hyper-parameters of the GPs. Without loss of generality, we assume Y to be univariate and the hidden layer to be composed of a single GP. The model is therefore as follows:

$$\begin{aligned} p\left(Y\middle|F^{(2)},\lambda\right) &= \mathcal{N}\left(Y\middle|F^{(2)},\lambda I\right) \\ p\left(F^{(2)}\middle|F^{(1)},\boldsymbol{\theta}^{(1)}\right) &= \mathcal{N}\left(F^{(2)}\middle|\mathbf{0},K\left(F^{(1)},\boldsymbol{\theta}^{(1)}\right)\right) \\ p\left(F^{(1)}\middle|X,\boldsymbol{\theta}^{(0)}\right) &= \mathcal{N}\left(F^{(1)}\middle|\mathbf{0},K\left(X,\boldsymbol{\theta}^{(0)}\right)\right) \end{aligned}$$

with λ, $\boldsymbol{\theta}^{(1)}$, and $\boldsymbol{\theta}^{(0)}$ fixed. In the model specification above, we denoted by $K\left(F^{(1)},\boldsymbol{\theta}^{(1)}\right)$ and $K\left(X,\boldsymbol{\theta}^{(0)}\right)$ the covariance matrices obtained by applying the covariance function with parameters $\boldsymbol{\theta}^{(1)}$, and $\boldsymbol{\theta}^{(0)}$ to all pairs of $F^{(1)}$ and X, respectively.

Given that the likelihood is Gaussian, it is possible to integrate out $F^{(2)}$ analytically

$$p\left(Y\middle|F^{(1)},\lambda,\boldsymbol{\theta}^{(1)}\right) = \int p\left(Y\middle|F^{(2)},\lambda\right) p\left(F^{(2)}\middle|F^{(1)},\boldsymbol{\theta}^{(1)}\right) dF^{(2)}$$

obtaining the more compact model specification:

$$\begin{aligned} p\left(Y\middle|F^{(1)},\lambda,\boldsymbol{\theta}^{(1)}\right) &= \mathcal{N}\left(Y\middle|\mathbf{0},K\left(F^{(1)},\boldsymbol{\theta}^{(1)}\right)+\lambda I\right) \\ p\left(F^{(1)}\middle|X,\boldsymbol{\theta}^{(0)}\right) &= \mathcal{N}\left(F^{(1)}\middle|\mathbf{0},K\left(X,\boldsymbol{\theta}^{(0)}\right)\right) \end{aligned}$$

For fixed hyper-parameters, these expressions reveal that the observations are distributed as in the standard GP regression case, with the only difference that the covariance is now parameterized by GP distributed random variables $F^{(1)}$. We can interpret these variables as some sort of hyper-parameters, and we can attempt to use standard MCMC methods to samples from their posterior.

In order to develop a sampler for all latent variables, we factorize their full posterior as follows:

$$p\left(F^{(2)},F^{(1)}\middle|Y,X,\lambda,\boldsymbol{\theta}^{(1)},\boldsymbol{\theta}^{(0)}\right) = p\left(F^{(2)}\middle|Y,F^{(1)},\lambda,\boldsymbol{\theta}^{(1)}\right) p\left(F^{(1)}\middle|Y,X,\lambda,\boldsymbol{\theta}^{(1)},\boldsymbol{\theta}^{(0)}\right)$$

which suggest a Gibbs sampling strategy to draw samples from the posterior where we iterate

1. sample from $p\left(F^{(1)}\middle|Y,X,\lambda,\boldsymbol{\theta}^{(1)},\boldsymbol{\theta}^{(0)}\right)$

2. sample from $p\left(F^{(2)}\middle|Y, F^{(1)}, \lambda, \boldsymbol{\theta}^{(1)}\right)$

Step 1. can be done by setting up a Markov chain with invariant distribution given by:

$$p\left(F^{(1)}\middle|Y, X, \lambda, \boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(0)}\right) \propto p\left(Y\middle|F^{(1)}, \lambda, \boldsymbol{\theta}^{(1)}\right) p\left(F^{(1)}\middle|X, \boldsymbol{\theta}^{(0)}\right)$$

We can interpret this as a GP model, where the likelihood now assumes a complex form because of the nonlinear way in which the likelihood depends on $F^{(1)}$. Because of this interpretation, we can attempt to use any of the samplers developed in the literature of GPs to obtain samples from the posterior over latent variables in GP models.

Step 2. can be done directly given that the posterior over $F^{(2)}$ is available in closed form and it is Gaussian:

$$p\left(F^{(2)}\middle|Y, F^{(1)}, \lambda, \boldsymbol{\theta}^{(1)}\right) = \mathcal{N}\left(F^{(2)}\middle|K^{(1)}\left(K^{(1)}+\lambda I\right)^{-1} Y, K^{(1)} - K^{(1)}\left(K^{(1)}+\lambda I\right)^{-1} K^{(1)}\right)$$

where we have defined

$$K^{(1)} := K\left(F^{(1)}, \boldsymbol{\theta}^{(1)}\right)$$

C. Derivation of the lower bound

For the sake of completeness, here is a detailed derivation of the lower bound that we use in variational inference to learn the posterior over $\mathbf{W}$ and optimize $\boldsymbol{\Theta}$, assuming $\boldsymbol{\Omega}$ fixed:

$$\begin{aligned}
\log[p(Y|X, \boldsymbol{\Omega}, \boldsymbol{\Theta})] &= \log\left[\int p(Y|X, \mathbf{W}, \boldsymbol{\Omega}, \boldsymbol{\Theta}) p(\mathbf{W}) d\mathbf{W}\right] \\
&= \log\left[\int \frac{p(Y|X, \mathbf{W}, \boldsymbol{\Omega}, \boldsymbol{\Theta}) p(\mathbf{W})}{q(\mathbf{W})} q(\mathbf{W}) d\mathbf{W}\right] \\
&= \log\left[\mathrm{E}_{q(\mathbf{W})} \frac{p(Y|X, \mathbf{W}, \boldsymbol{\Omega}, \boldsymbol{\Theta}) p(\mathbf{W})}{q(\mathbf{W})}\right] \\
&\geq \mathrm{E}_{q(\mathbf{W})}\left(\log\left[\frac{p(Y|X, \mathbf{W}, \boldsymbol{\Omega}, \boldsymbol{\Theta}) p(\mathbf{W})}{q(\mathbf{W})}\right]\right) \\
&= \mathrm{E}_{q(\mathbf{W})}\left(\log[p(Y|X, \mathbf{W}, \boldsymbol{\Omega}, \boldsymbol{\Theta})]\right) + \mathrm{E}_{q(\mathbf{W})}\left(\log\left[\frac{p(\mathbf{W})}{q(\mathbf{W})}\right]\right) \\
&= \mathrm{E}_{q(\mathbf{W})}\left(\log[p(Y|X, \mathbf{W}, \boldsymbol{\Omega}, \boldsymbol{\Theta})]\right) - \mathrm{DKL}[q(\mathbf{W})||p(\mathbf{W})]
\end{aligned}$$

D. Learning Ω variationally

Defining $\boldsymbol{\Psi} = \{\mathbf{W}, \boldsymbol{\Omega}\}$, we can attempt to employ variational inference to treat the spectral frequencies $\boldsymbol{\Omega}$ variationally as well as $\mathbf{W}$. In this case, the detailed derivation of the lower bound is as follows:

$$\begin{aligned}
\log\left[p(Y|X, \boldsymbol{\Theta})\right] &= \log\left[\int p(Y|X, \boldsymbol{\Psi}, \boldsymbol{\Theta}) p(\boldsymbol{\Psi}|\boldsymbol{\Theta}) d\boldsymbol{\Psi}\right] \\
&= \log\left[\int \frac{p(Y|X, \boldsymbol{\Psi}, \boldsymbol{\Theta}) p(\boldsymbol{\Psi}|\boldsymbol{\Theta})}{q(\boldsymbol{\Psi})} q(\boldsymbol{\Psi}) d\boldsymbol{\Psi}\right] \\
&= \log\left[\mathrm{E}_{q(\boldsymbol{\Psi})} \frac{p(Y|X, \boldsymbol{\Psi}, \boldsymbol{\Theta}) p(\boldsymbol{\Psi}|\boldsymbol{\Theta})}{q(\boldsymbol{\Psi})}\right] \\
&\geq \mathrm{E}_{q(\boldsymbol{\Psi})}\left(\log\left[\frac{p(Y|X, \boldsymbol{\Psi}, \boldsymbol{\Theta}) p(\boldsymbol{\Psi}|\boldsymbol{\Theta})}{q(\boldsymbol{\Psi})}\right]\right) \\
&= \mathrm{E}_{q(\boldsymbol{\Psi})}\left(\log[p(Y|X, \boldsymbol{\Psi}, \boldsymbol{\Theta})]\right) + \mathrm{E}_{q(\boldsymbol{\Psi})}\left(\log\left[\frac{p(\boldsymbol{\Psi}|\boldsymbol{\Theta})}{q(\boldsymbol{\Psi})}\right]\right) \\
&= \mathrm{E}_{q(\boldsymbol{\Psi})}\left(\log[p(Y|X, \boldsymbol{\Psi}, \boldsymbol{\Theta})]\right) - \mathrm{DKL}[q(\boldsymbol{\Psi})||p(\boldsymbol{\Psi}|\boldsymbol{\Theta})]
\end{aligned}$$

Again, assuming a factorized prior over all weights across layers

$$p(\mathbf{\Psi}|\boldsymbol{\theta}) = \prod_{l=0}^{N_{\text{h}}-1} p(\Omega^{(l)}|\boldsymbol{\theta}^{(l)})p(W^{(l)}) = \prod_{ijl} q\left(\Omega_{ij}^{(l)}\right) \prod_{ijl} q\left(W_{ij}^{(l)}\right), \tag{1}$$

we optimize the variational lower bound using variational inference following the mini-batch approach with the reparameterization trick explained in the main paper. The variational parameters then become the mean and the variance of each of the approximating factors

$$q\left(W_{ij}^{(l)}\right) = \mathcal{N}\left(m_{ij}^{(l)}, (s^2)_{ij}^{(l)}\right), \tag{2}$$

$$q\left(\Omega_{ij}^{(l)}\right) = \mathcal{N}\left(\mu_{ij}^{(l)}, (\beta^2)_{ij}^{(l)}\right), \tag{3}$$

and we optimize the lower bound with respect to the variational parameters $m_{ij}^{(l)}, (s^2)_{ij}^{(l)}, \mu_{ij}^{(l)}, (\beta^2)_{ij}^{(l)}$.

E. Expression for the DKL divergence between Gaussians

Given $p_1(x) = \mathcal{N}(\mu_1, \sigma_1^2)$ and $p_2(x) = \mathcal{N}(\mu_2, \sigma_2^2)$, the KL divergence between the two is:

$$\text{DKL}\left(p_1(x)\|p_2(x)\right) = \frac{1}{2}\left[\log\left(\frac{\sigma_2^2}{\sigma_1^2}\right) - 1 + \frac{\sigma_1^2}{\sigma_2^2} + \frac{(\mu_1 - \mu_2)^2}{\sigma_2^2}\right]$$

F. Distributed Implementation

Our model is easily amenable to a distributed implementation using asynchronous distributed stochastic gradient descent (Chilimbi et al., *USENIX*, 2014). Our distributed setting, based on TensorFlow, includes one or more *Parameter servers* (PS), and a number of *Workers*. The latter proceed asynchronously using randomly selected batches of data: they fetch fresh model parameters from the PS, compute the gradients of the lower bound with respect to these parameters, and push those gradients back to the PS, which update the model accordingly. Given that workers compute gradients and send updates to PS asynchronously, the discrepancy between the model used to compute gradients and the model actually updated can degrade training quality. This is exacerbated by a large number of asynchronous workers, as noted in Chen et al., *arXiv 1604.00981*, (2016).

We focus our experiments on the MNIST dataset, and study how training time and error rates evolve with the number of workers introduced in our system. The parameters for the model are identical to those reported for the previous experiments, except that we fix the number of Monte Carlo samples to one throughout.

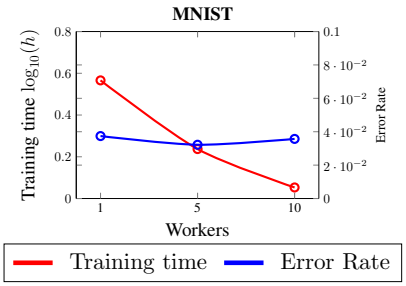

Figure 5. Comparison of training time and error rate for asynchronous DGP-RFF with 2 PS, and 1, 5 and 10 workers.

We report the results in Figure 5. The training time decreases in proportion to the number of workers, albeit sub-linearly, whereas the error rate remains almost the same across different numbers of workers. When using a single PS, we noticed that increasing the number of workers led to lower gains in speed of execution, and the behavior of the error rate over iterations was more erratic than in the case of two PS. We attribute these effects to the greater overheads on the communication cost associated with the configuration with a single PS, and the more involved handling of the coordination of multiple workers. The work in Chen et al., *arXiv 1604.00981*, (2016) corroborates our findings, and motivates efforts in the direction of alleviating this issue.