Deep Gaussian Processes for Regression using Approximate Expectation Propagation: Supplementary material

Thang D. Bui[1] TDB40@CAM.AC.UK
José Miguel Hernández-Lobato[2] JMH@SEAS.HARVARD.EDU
Daniel Hernández-Lobato[3] DANIEL.HERNANDEZ@UAM.ES
Yingzhen Li[1] YL494@CAM.AC.UK
Richard E. Turner[1] RET26@CAM.AC.UK
[1]University of Cambridge, [2]Harvard University, [3]Universidad Autónoma de Madrid

1. Approximate predictive distribution

Given the approximate posterior and a new test input x^*, we wish to make a prediction about the test output y^*. That is to find $p(y^*|x^*, \mathbf{X}, \mathbf{Y}) \approx \int \mathrm{d}\mathbf{u}\, p(y^*|x^*, \mathbf{u})\, q(\mathbf{u}|\mathbf{X}, \mathbf{Y})$. This predictive distribution is not analytically tractable, but fortunately, we can approximate it by a Gaussian in a similar fashion to the method described in the main text. That is, a single forward pass is performed, in which each layer takes in a Gaussian distribution over the input, incorporates the approximate posterior of the inducing outputs and approximates the output distribution by a Gaussian. An alternative to obtain the prediction is to forward sample from the model, but we do not use this approach in the experiments.

2. Extra experimental results

2.1. Regression

Due to the page limitation of the main text, we include here several figures and tables showing the full experimental results and analyses from the regression experiments on 10 UCI datasets. Note that the results for DGPs reported here could be improved further by increasing the number of pseudo datapoints. We choose 50 and 100 pseudo datapoints (or 100 and 200 for the big datasets) so that the training time and prediction time are comparable across all methods. Next we show the full results for the implemented methods and the their average rank across all train/test splits.

- Figures 1 and 2 show the full MLL results for all methods and all datasets. Part of these results have been included in the main text. These figures show that DGPs with our approximation scheme is superior as measured by the MLL metric, obtaining the top spot in the average ranking table.
- Figures 3 and 4 show the full RMSE results for all methods. Surprisingly, though not doing well on the MLL metric, i.e. providing inaccurate predictive uncertainty, BNN-SGLD with one and two layers are very good at predicting the mean of the test set. DGPs, on average, rival or perform better than this approximate sampling scheme and other methods.
- Figures 5 and 6 show the subset of the MLL results above, for GP architectures, and their average ranking. This again demonstrate that DGPs are more flexible than GPs, hence always obtain better predictive performance. The only exception is the network with a one dimensional hidden layer or a warped GP which performs poorly relative to other architectures.
- Similarly, Figures 7 and 8 show evidence that increasing the number of layers and hidden dimensions helps improving the accuracy of the predictions.
- We include a similar analysis for approximate inference methods for BNNs in Figures 9, 10, 11 and 12. This set of results demonstrates that VI(KW) and SGLD with two hidden layers provide good performance on the test sets, outperforming other methods in shallower architectures. HMC with one hidden layer performs well overall, but its running time is much larger compared to other methods. Other deterministic approximations [VI(G), PBP and Dropout] perform poorly overall.

Tables 3 and 4 show the average test log-likelihood and error respectively for all datasets. The best deterministic method for each dataset is bolded, the best method overall (deterministic and sampling) is underlined and emphasised in italic. The average ranks of the methods across the 10 datasets are also included.

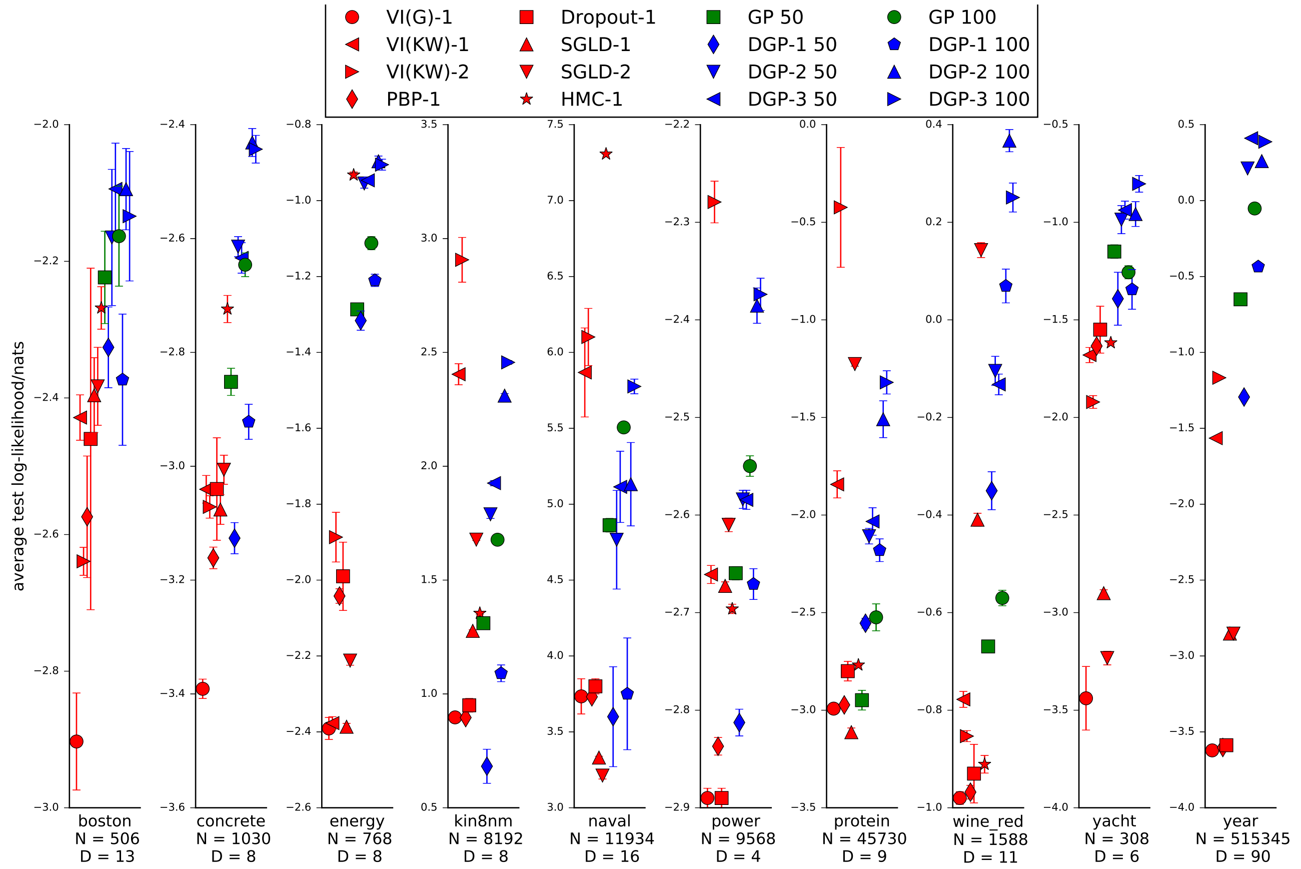

Figure 1. Average test log likelihood for all methods

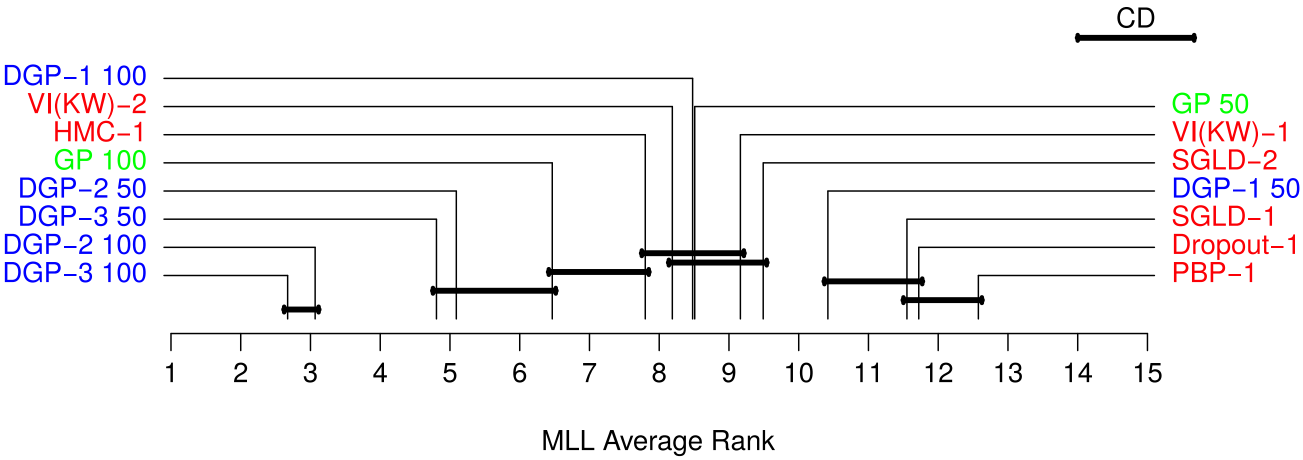

Figure 2. The average rank based on the test MLL of all methods across the datasets and their train/test splits, generated based on (Demšar, 2006). See the main text for more details.

2.2. Binary and multiclass classification

We test our approximate inference scheme for DGPs with non-Gaussian noise models. However, as shown in Tables 1 and 2, DGPs often obtain a marginal gain over GPs, as compared to some substantial improvement in the regression experiments above. We speculate that this is due to our current initialisation strategy and our diagonal Gaussian approximation at last layer for multiclass classification. We will follow this up in future work.

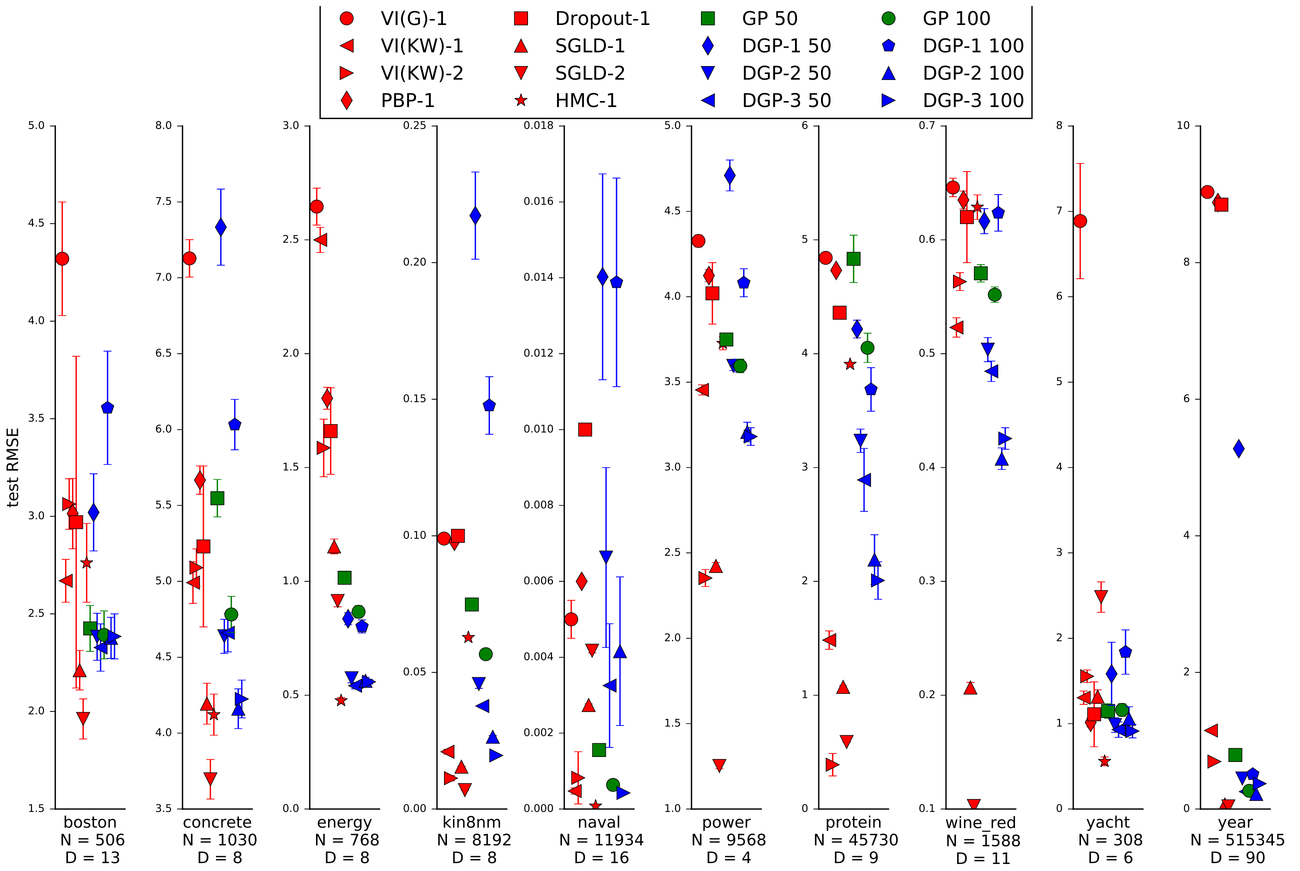

Figure 3. Average test RMSE for all methods

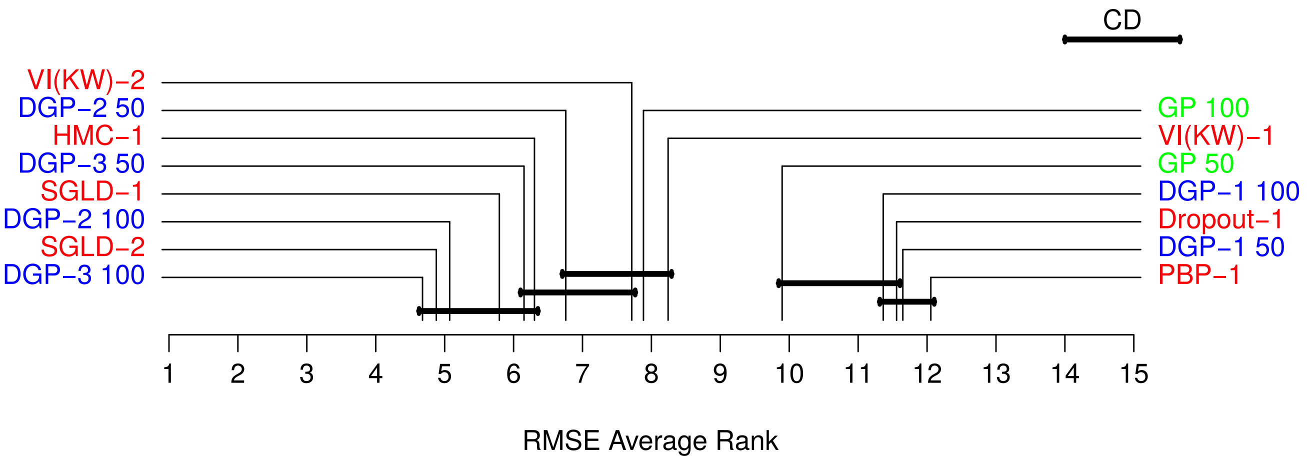

Figure 4. The average rank based on the test RMSE of all methods across the datasets and their train/test splits, generated based on (Demšar, 2006). See the main text for more details.

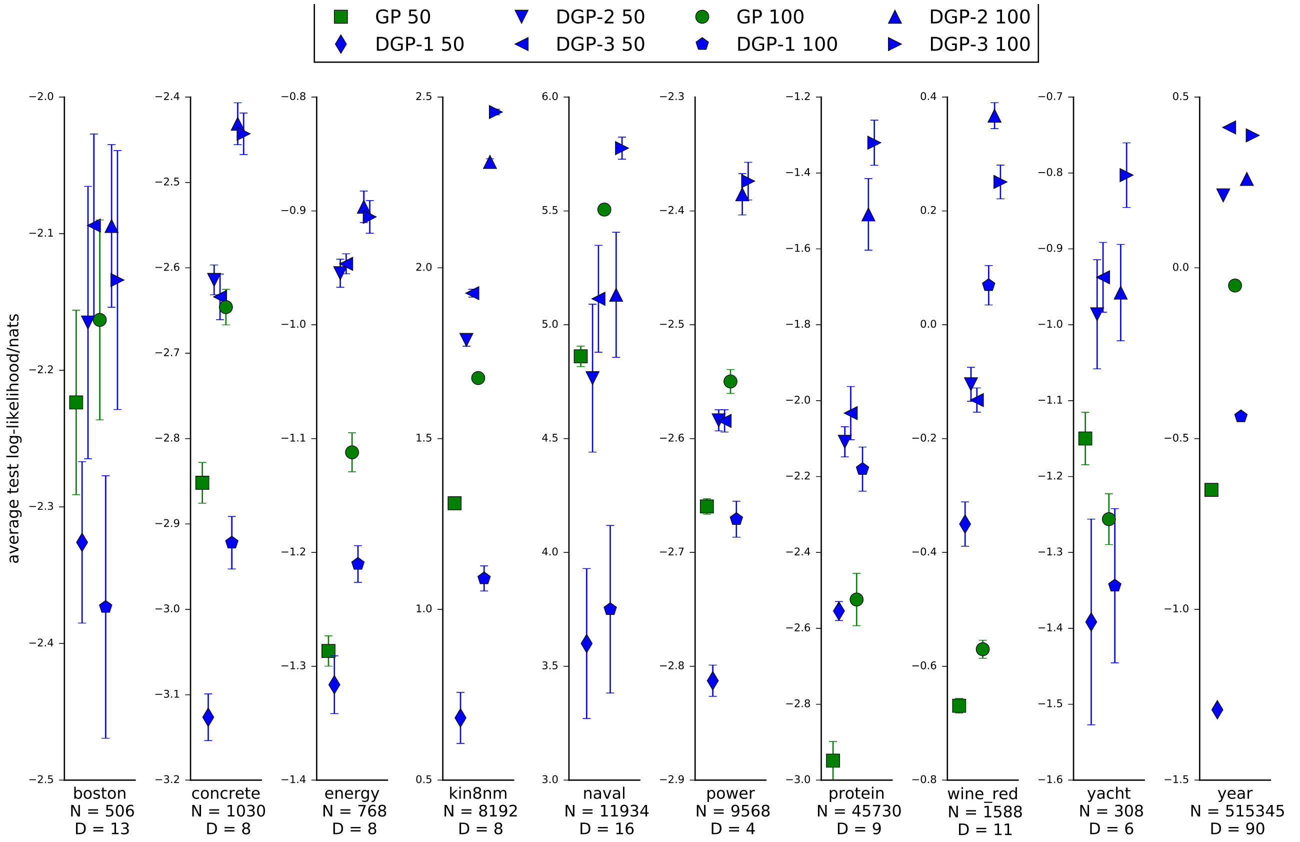

Figure 5. Average test log likelihood for GP methods

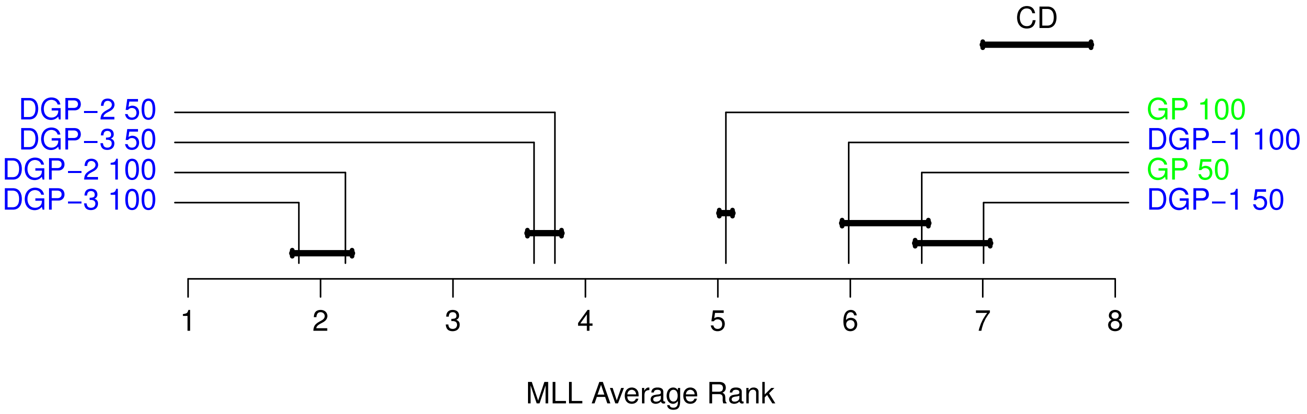

Figure 6. The average rank based on the test MLL for GP/DGP models across the datasets and their train/test splits, generated based on (Demšar, 2006). See the main text for more details.

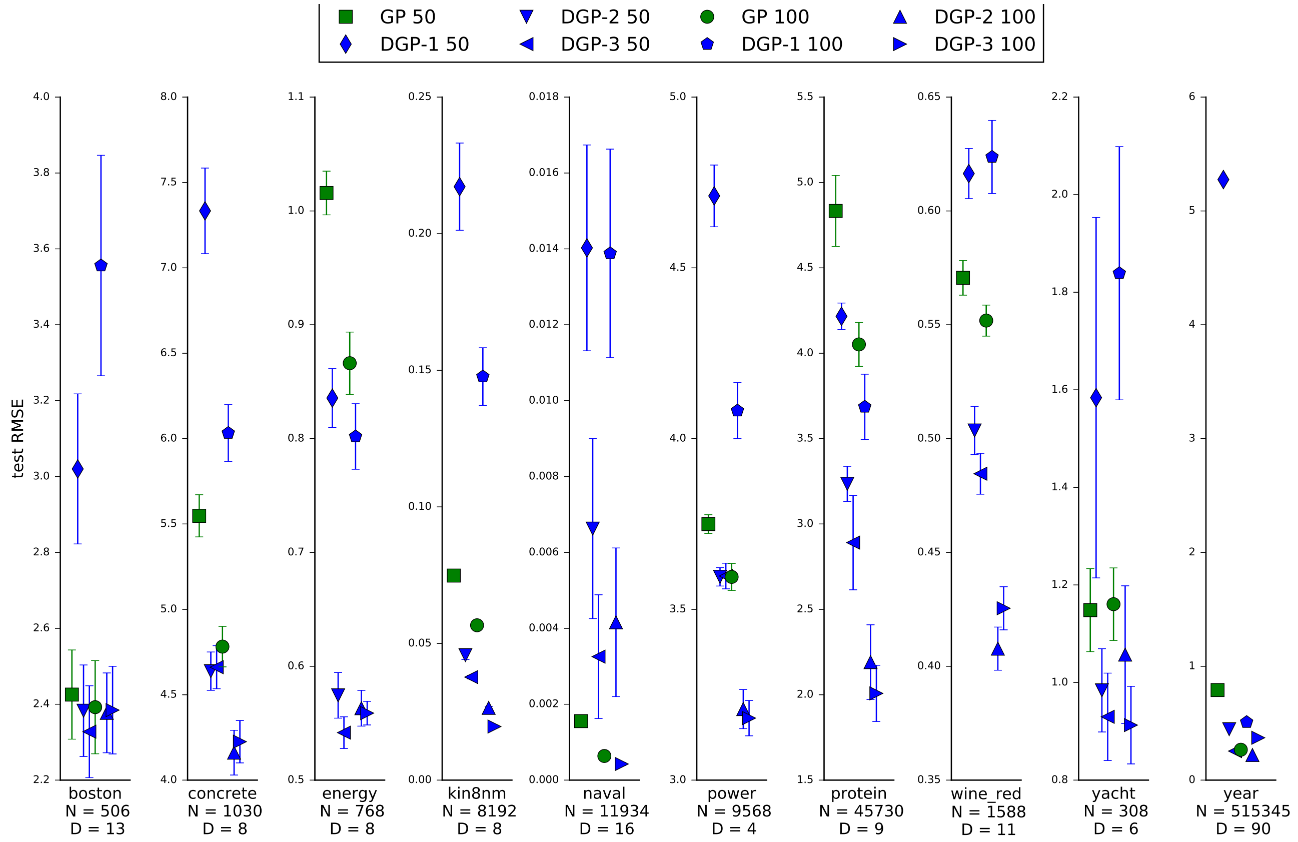

Figure 7. Average test RMSE for GP methods

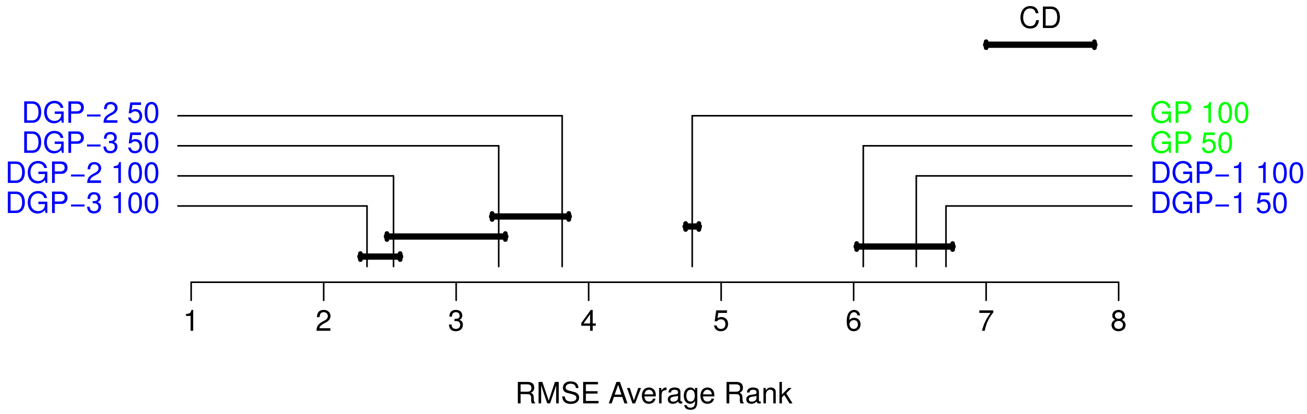

Figure 8. The average rank based on the test RMSE for GP/DGP models across the datasets and their train/test splits, generated based on (Demšar, 2006). See the main text for more details.

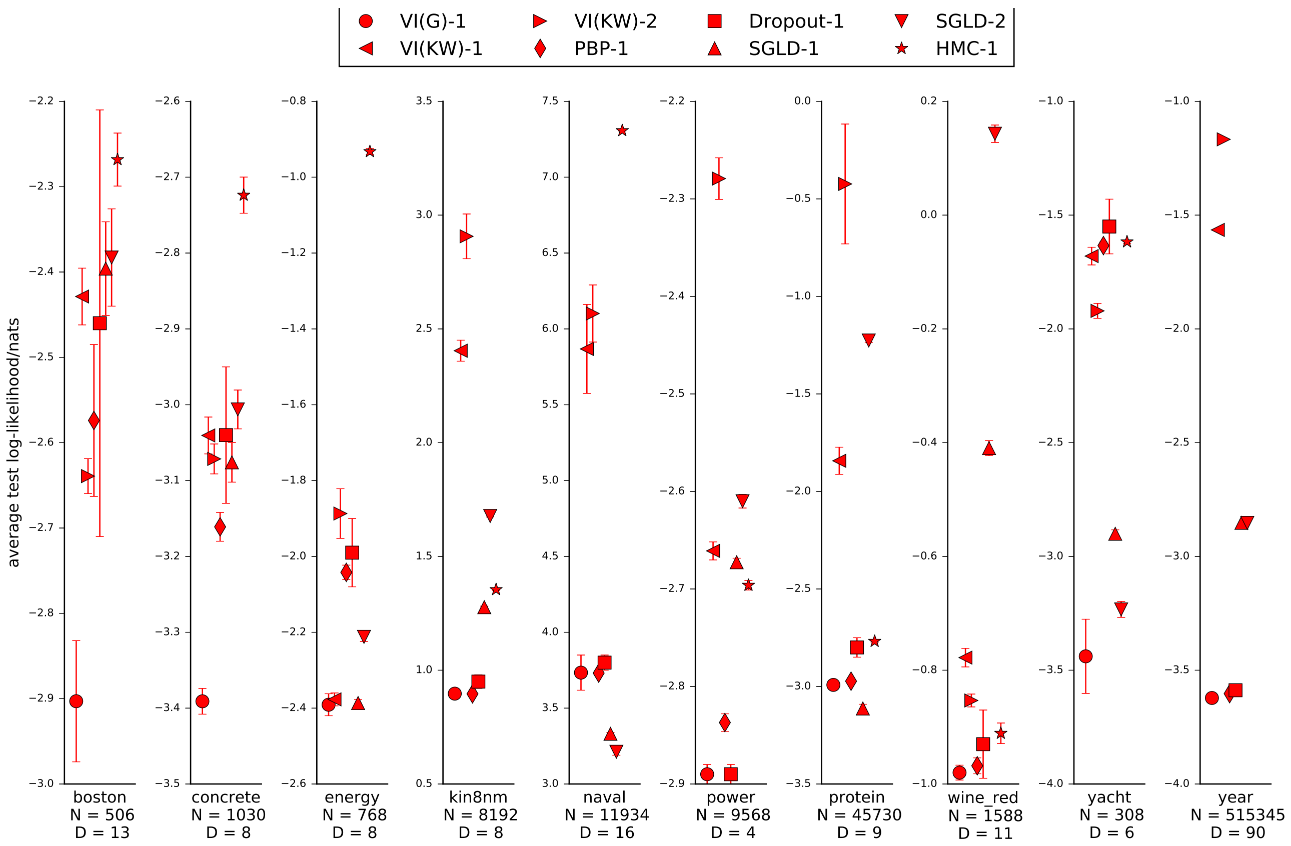

Figure 9. Average test log likelihood for methods with BNNs

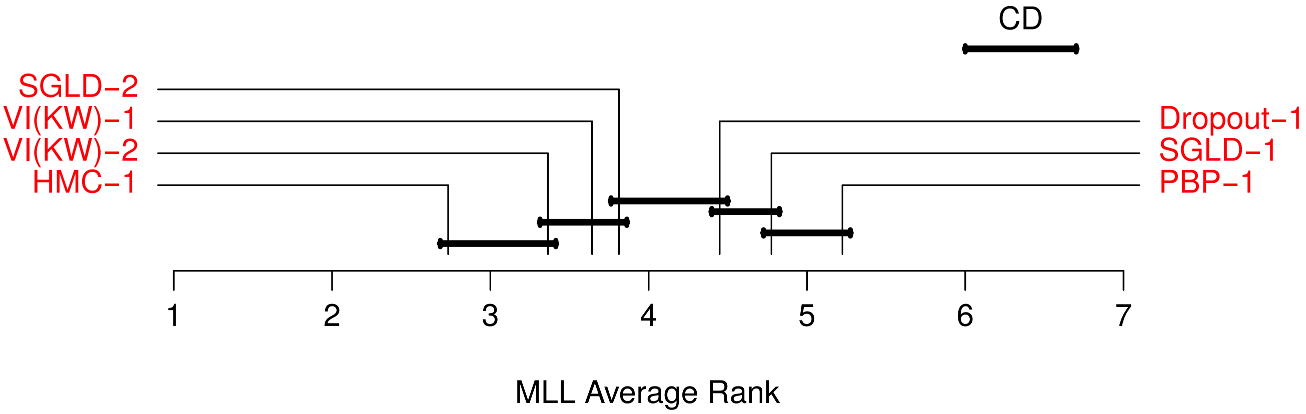

Figure 10. The average rank based on the test MLL for methods on BNNs across the datasets and their train/test splits, generated based on (Demšar, 2006). See the main text for more details.

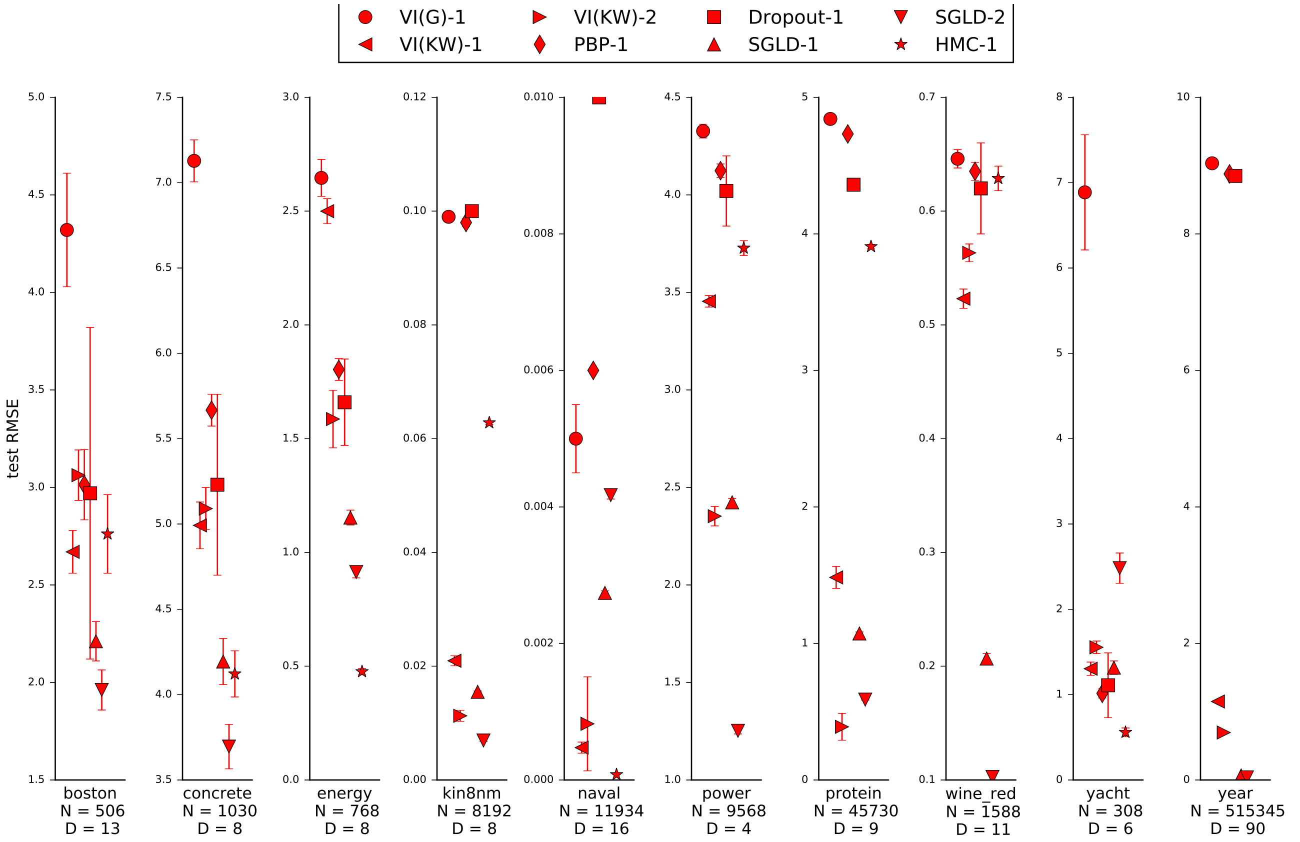

Figure 11. Average test RMSE for methods with BNNs

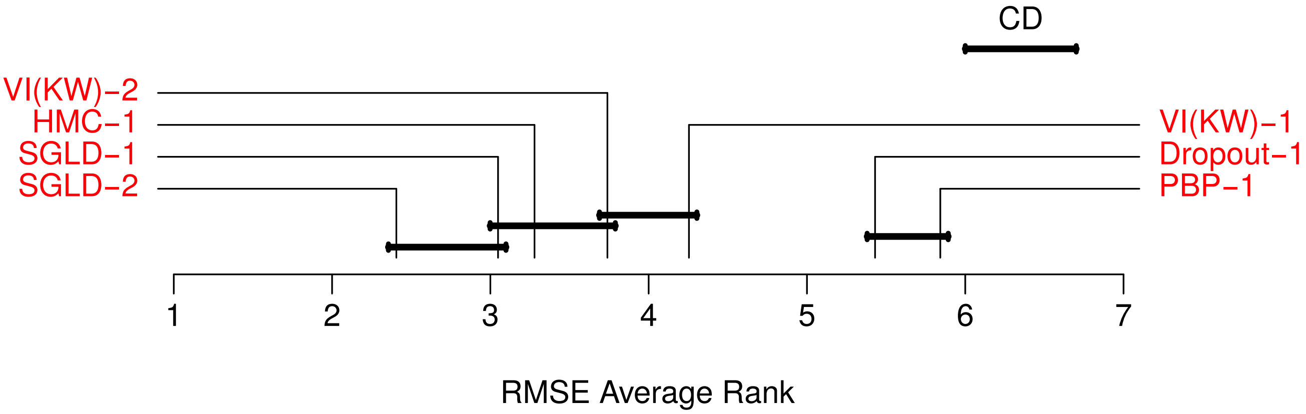

Figure 12. The average rank based on the test RMSE for methods on BNNs across the datasets and their train/test splits, generated based on (Demšar, 2006). See the main text for more details.

Table 1. Binary cla. experiment: Average test log-likelihood/nats

Dataset	GP D-1	DGP D-1-1	DGP D-2-1	DGP D-3-1
australian	**-0.51±0.01**	-0.51±0.02	-0.51±0.02	-0.53±0.02
breast	-0.05±0.01	-0.04±0.01	**-0.04±0.01**	-0.04±0.01
crabs	**-0.03±0.01**	-0.10±0.05	-0.03±0.01	-0.03±0.01
ionoshere	-0.17±0.02	-0.17±0.03	-0.16±0.03	**-0.16±0.02**
pima	-0.40±0.01	-0.39±0.01	-0.40±0.02	**-0.39±0.01**
sonar	-0.32±0.03	**-0.29±0.03**	-0.30±0.03	-0.31±0.03

Table 2. Multiclass experiment: Average test log-likelihood/nats

Dataset	N	D	K	GP D-K	DGP D-1-K	GP D-2-K	DGP D-3-K
glass	214	9	6	-0.79±0.02	**-0.71±0.02**	-0.72±0.02	-0.71±0.02
new-thyroid	215	5	3	-0.05±0.01	-0.05±0.01	-0.05±0.02	**-0.04±0.01**
svmguide2	319	20	3	-0.54±0.02	-0.53±0.02	-0.52±0.02	**-0.51±0.02**
wine	178	13	3	-0.10±0.01	**-0.07±0.01**	-0.07±0.01	-0.07±0.01

Table 3. Regression experiment: Average test log likelihood/nats

Dataset	N	D	VI(G)-1	VI(KW)-1	VI(KW)-2	PBP-1	Dropout-1	SGLD-1	SGLD-2	HMC-1	GP 50	DGP-1 50	DGP-2 50	DGP-3 50	GP 100	DGP-1 100	DGP-2 100	DGP-3 100
boston	506	13	-2.90±0.07	-2.43±0.03	-2.64±0.02	-2.57±0.09	-2.46±0.25	-2.40±0.05	-2.38±0.06	-2.27±0.03	-2.22±0.07	-2.33±0.06	-2.17±0.10	***-2.09±0.07***	-2.16±0.07	-2.37±0.10	-2.09±0.06	-2.13±0.09
concrete	1030	8	-3.39±0.02	-3.04±0.02	-3.07±0.02	-3.16±0.02	-3.04±0.09	-3.08±0.03	-3.01±0.03	-2.72±0.02	-2.85±0.02	-3.13±0.03	-2.61±0.02	-2.63±0.03	-2.65±0.02	-2.92±0.03	***-2.43±0.02***	-2.44±0.02
energy	768	8	-2.39±0.03	-2.38±0.02	-1.89±0.07	-2.04±0.02	-1.99±0.09	-2.39±0.01	-2.21±0.01	-0.93±0.01	-1.29±0.01	-1.32±0.03	-0.95±0.01	-0.95±0.01	-1.11±0.02	-1.21±0.02	***-0.90±0.01***	-0.91±0.01
kin8nm	8192	8	0.90±0.01	2.40±0.05	***2.91±0.10***	0.90±0.01	0.95±0.03	1.28±0.00	1.68±0.00	1.35±0.00	1.31±0.01	0.68±0.07	1.79±0.02	1.93±0.01	1.68±0.01	1.09±0.04	2.31±0.01	2.46±0.01
naval	11934	16	3.73±0.12	5.87±0.29	**6.10±0.19**	3.73±0.01	3.80±0.05	3.33±0.01	3.21±0.02	*7.31±0.00*	4.86±0.04	3.60±0.33	4.77±0.32	5.11±0.23	5.51±0.03	3.75±0.37	5.13±0.27	5.78±0.05
power	9568	4	-2.89±0.01	-2.66±0.01	***-2.28±0.02***	-2.84±0.01	-2.89±0.01	-2.67±0.00	-2.61±0.01	-2.70±0.00	-2.66±0.01	-2.81±0.01	-2.58±0.01	-2.58±0.01	-2.55±0.01	-2.67±0.02	-2.39±0.02	-2.37±0.02
protein	45730	9	-2.99±0.01	-1.84±0.07	***-0.42±0.31***	-2.97±0.00	-2.80±0.05	-3.11±0.02	-1.23±0.01	-2.77±0.00	-2.95±0.05	-2.55±0.03	-2.11±0.04	-2.03±0.07	-2.52±0.07	-2.18±0.06	-1.51±0.09	-1.32±0.06
red wine	1588	11	-0.98±0.01	-0.78±0.02	-0.85±0.01	-0.97±0.01	-0.93±0.06	-0.41±0.01	0.14±0.02	-0.91±0.02	-0.67±0.01	-0.35±0.04	-0.10±0.03	-0.13±0.02	-0.57±0.02	0.07±0.03	***0.37±0.02***	0.25±0.03
yacht	308	6	-3.44±0.16	-1.68±0.04	-1.92±0.03	-1.63±0.02	-1.55±0.12	-2.90±0.02	-3.23±0.03	-1.62±0.01	-1.15±0.03	-1.39±0.14	-0.99±0.07	-0.94±0.05	-1.26±0.03	-1.34±0.10	-0.96±0.06	***-0.80±0.04***
year	515345	90	-3.62±NA	-1.56±NA	-1.17±NA	-3.60±NA	-3.59±NA	-2.85±NA	-2.85±NA	NA±NA	-0.65±NA	-1.29±NA	0.21±NA	***0.41***±NA	-0.05±NA	-0.44±NA	0.26±NA	0.39±NA
Average Rank			15.10±0.39	9.00±1.18	7.50±1.70	13.70±0.40	12.10±0.64	12.50±0.75	9.40±1.42	8.80±1.38	8.20±0.69	10.80±0.95	5.30±0.51	4.20±0.66	6.10±0.57	8.20±0.72	2.80±0.49	**2.30±0.25**

Table 4. Regression experiment: Test root mean square error

Dataset	N	D	VI(G)-1	VI(KW)-1	VI(KW)-2	PBP-1	Dropout-1	SGLD-1	SGLD-2	HMC-1	GP 50	DGP-1 50	DGP-2 50	DGP-3 50	GP 100	DGP-1 100	DGP-2 100	DGP-3 100
boston	506	13	4.32±0.29	2.67±0.11	3.06±0.13	3.01±0.18	2.97±0.85	2.21±0.10	*1.96±0.10*	2.76±0.20	2.43±0.12	3.02±0.20	2.38±0.12	**2.33±0.12**	2.39±0.12	3.56±0.29	2.38±0.11	2.38±0.12
concrete	1030	8	7.13±0.12	4.99±0.14	5.09±0.12	5.67±0.09	5.23±0.53	4.19±0.13	*3.70±0.13*	4.12±0.14	5.55±0.12	7.33±0.25	4.64±0.11	4.66±0.13	4.78±0.12	6.03±0.17	**4.16±0.13**	4.23±0.12
energy	768	8	2.65±0.08	2.50±0.06	1.59±0.13	1.80±0.05	1.66±0.19	1.15±0.03	0.91±0.03	*0.48±0.01*	1.02±0.02	0.84±0.03	0.57±0.02	**0.54±0.01**	0.87±0.03	0.80±0.03	0.56±0.02	0.56±0.01
kin8nm	8192	8	0.10±0.00	0.02±0.00	**0.01±0.00**	0.10±0.00	0.10±0.00	0.02±0.00	*0.01±0.00*	0.06±0.00	0.07±0.00	0.22±0.02	0.05±0.00	0.04±0.00	0.06±0.00	0.15±0.01	0.03±0.00	0.02±0.00
naval	11934	16	0.01±0.00	0.00±0.00	0.00±0.00	0.01±0.00	0.01±0.00	0.00±0.00	0.00±0.00	*0.00±0.00*	0.00±0.00	0.01±0.00	0.01±0.00	0.00±0.00	0.00±0.00	0.01±0.00	0.00±0.00	**0.00±0.00**
power	9568	4	4.33±0.04	3.45±0.03	**2.35±0.05**	4.12±0.03	4.02±0.18	2.42±0.02	*1.25±0.02*	3.73±0.04	3.75±0.03	4.71±0.09	3.60±0.03	3.60±0.04	3.60±0.04	4.08±0.08	3.21±0.06	3.18±0.05
protein	45730	9	4.84±0.03	1.48±0.08	***0.39±0.10***	4.73±0.01	4.36±0.04	1.07±0.01	0.59±0.00	3.91±0.02	4.83±0.21	4.22±0.08	3.24±0.10	2.89±0.28	4.05±0.13	3.69±0.19	2.19±0.22	2.01±0.16
red wine	1588	11	0.65±0.01	0.52±0.01	0.56±0.01	0.64±0.01	0.62±0.04	0.21±0.00	*0.10±0.00*	0.63±0.01	0.57±0.01	0.62±0.01	0.50±0.01	0.48±0.01	0.55±0.01	0.62±0.02	**0.41±0.01**	0.43±0.01
yacht	308	6	6.89±0.67	1.30±0.08	1.55±0.07	1.01±0.05	1.11±0.38	1.32±0.08	2.48±0.18	*0.56±0.05*	1.15±0.09	1.58±0.37	0.98±0.09	0.93±0.09	1.16±0.07	1.84±0.26	1.06±0.14	**0.91±0.08**
year	515345	90	9.03±NA	1.15±NA	0.70±NA	8.88±NA	8.85±NA	0.07±NA	*0.04*±NA	NA±NA	0.79±NA	5.28±NA	0.45±NA	0.26±NA	0.27±NA	0.51±NA	**0.22**±NA	0.37±NA
Average Rank			14.90±0.50	7.90±1.09	7.60±1.42	12.50±0.85	12.00±0.62	4.80±1.08	4.20±1.55	7.50±1.72	10.10±0.74	13.20±0.88	7.00±0.76	5.50±0.72	7.60±0.60	12.20±0.99	4.90±0.57	**4.10±0.43**

2.3. Comparison to the Nested Variational approach

Recent work by Hensman and Lawrence (2014) proposed using a nested variational scheme (Nested VI) to perform inference in DGPs. There is an error in the final bound in (Hensman & Lawrence, 2014), and hence a corresponding error in the implementation provided here https://github.com/SheffieldML/deepGPy [visited on 25/05/2016]. However, we still use this version of the code to run the comparion below.

We compare the proposed method using approximate EP (AEP) against Nested VI on 4 UCI regression datasets using Deep GPs with two GP layers and a one-dimensional hidden layer. We vary the number of pseudo points M and measure the performances using two metrics, RMSE and NLL as described in the main text. The results are included in figs. 13 and 14, demonstrating that the proposed approach, AEP outperforms Nested VI by a large margin. We also note that Nested VI often consistently performs poorly and does not improve as more pseudo points are added.

3. EP and SEP

In this section, we summarise the EP and SEP iterative procedures. The EP algorithm is often mistaken to be optimising $\mathrm{KL}(p(\mathbf{u}|\mathbf{X},\mathbf{y})||q(\mathbf{u}))$; however, this objective function is intractable. Instead, EP updates one approximate factor at a time by the following procedure: 1. remove the factor $\tilde{t}_n(\mathbf{u})$ to form the leave-one-out or cavity distribution $q^{\setminus n}(\mathbf{u}) \propto q(\mathbf{u})/\tilde{t}_n(\mathbf{u})$, 2. minimise $\mathrm{KL}(q^{\setminus n}(\mathbf{u})p(y_n|\mathbf{u},\mathbf{X}_n)||q(\mathbf{u}))$, resulting in a new approximate factor $\tilde{t}_n^{\text{new}}(\mathbf{u})$ which can be 3. combined with the cavity to form the new approximate posterior. This procedure is iteratively performed for each datapoint, and often requires several passes through the training set for convergence. One disadvantage of the EP algorithm is the need to store the approximate factors in memory, which costs $\mathcal{O}(NM^2)$.

To sidestep this expensive memory requirement, the SEP algorithm proposes tying the approximate data factors, that is to make some or all factors the same. The simplest case is $q(\mathbf{u}) \propto p(\mathbf{u})g(\mathbf{u})^N$ where $g(\mathbf{u})$ is the *average* data factor. The SEP algorithm, similar to EP, involves iteratively finding the new approximate factor $g_{\text{new}}(\mathbf{u})$, as follows: 1. remove the factor $\tilde{g}(\mathbf{u})$ to form the leave-one-out or cavity distribution $q^{\setminus 1}(\mathbf{u}) \propto q(\mathbf{u})/\tilde{g}(\mathbf{u})$, 2. minimise $\mathrm{KL}(q^{\setminus 1}(\mathbf{u})p(y_n|\mathbf{u},\mathbf{X}_n)||q(\mathbf{u}))$, resulting in a new approximate factor $\tilde{g}_{\text{new}}(\mathbf{u})$ which can be 3. combined with the cavity to form the new approximate posterior, and in addition to EP, 4. perform an explicit update to the *average* factor $g(\mathbf{u})$: $g(\mathbf{u}) \leftarrow g^{1-\beta}(\mathbf{u})g^{\beta}_{\text{new}}(\mathbf{u})$, where β is a small learning rate.

4. EP/SEP moment matching step

We have proposed using the EP approximate marginal likelihood for direct optimisation of the approximate posterior over the pseudo datapoints and the hyperparameters. An alternative is to run SEP/EP to obtain the approximate posterior, and once this is done, obtain the approximate marginal likelihood for hyperparameter tuning and repeat.

As we use Gaussian EP/SEP, the deletion, the update step and the explicit update step in the case of SEP are straightforward. The moment matching step is equivalent to the following updates to the mean and covariance of the approximate posterior:

$$
\begin{aligned}
\mathbf{m} &= \mathbf{m}^{\setminus 1} + \mathbf{V}^{\setminus 1}\frac{\mathrm{d}\log\mathcal{Z}}{\mathrm{d}\mathbf{m}^{\setminus 1}} \\
\mathbf{V} &= \mathbf{V}^{\setminus 1} - \mathbf{V}^{\setminus 1}\left[\frac{\mathrm{d}\log\mathcal{Z}}{\mathrm{d}\mathbf{m}^{\setminus 1}}\left(\frac{\mathrm{d}\log\mathcal{Z}}{\mathrm{d}\mathbf{m}^{\setminus 1}}\right)^{\intercal} - 2\frac{\mathrm{d}\log\mathcal{Z}}{\mathrm{d}\mathbf{V}^{\setminus 1}}\right]\mathbf{V}^{\setminus 1},
\end{aligned}
$$

where $q^{\setminus 1}(\mathbf{u}) = \mathcal{N}(\mathbf{u};\mathbf{m}^{\setminus 1},\mathbf{V}^{\setminus 1})$ is the cavity distribution, obtained by the deletion step.

The inference scheme therefore reduces to evaluating the normalising constant $\mathcal{Z}$ and its gradient. Fortunately, we can approximately compute $\log\mathcal{Z}$ and its gradients using the probabilistic propagation algorithm, in exactly the same way as discussed in the main text.

5. Computing the gradients of $\log\mathcal{Z}$

Let m_l and v_l be the mean and variance of the output Gaussian at the l-th layer in the forward propagation step, as we have shown in the main text,

$$
\begin{aligned}
m_l &= \psi_{l,1}\mathbf{A}_l && (1) \\
v_l &= \sigma_l^2 + \psi_{l,0} + \mathrm{tr}\left(\mathbf{B}_l\psi_{l,2}\right) - m_l^2 && (2)
\end{aligned}
$$

where

$$
\begin{aligned}
\psi_{l,0} &= \mathrm{E}_{q(h_1)}[K_{h_l,h_l}] && (3) \\
\psi_{l,1} &= \mathrm{E}_{q(h_{l-1})}[\mathbf{K}_{h_l,\mathbf{u}_l}] && (4) \\
\psi_{l,1} &= \mathrm{E}_{q(h_{l-1})}[\mathbf{K}_{\mathbf{u}_l,h_l}\mathbf{K}_{h_l,\mathbf{u}_l}] && (5) \\
\mathbf{A}_l &= \mathbf{K}^{-1}_{\mathbf{u}_l,\mathbf{u}_l}\mathbf{m}_l^{\setminus 1} && (6) \\
\mathbf{B}_l &= \mathbf{K}^{-1}_{\mathbf{u}_l,\mathbf{u}_l}(\mathbf{V}_l^{\setminus 1} + \mathbf{m}_l^{\setminus 1}\mathbf{m}_l^{\setminus 1,\mathrm{T}})\mathbf{K}^{-1}_{\mathbf{u}_l,\mathbf{u}_l} - \mathbf{K}^{-1}_{\mathbf{u}_l,\mathbf{u}_l} && (7)
\end{aligned}
$$

In the forward propagation step, we need to compute the gradients of m_l and v_l w.r.t. α_l, the parameters of the model and m_{l-1} and v_{l-1}, the mean and variance of the distribution over the input. Let $\beta_l = \{\alpha_l, m_{l-1}, v_{l-1}\}$ As $\mathbf{A}_l$ and $\mathbf{B}_l$ are shared between datapoints, one trick to reduce the computation required for each datapoint is to compute the gradients w.r.t. $\mathbf{A}$ and $\mathbf{B}$ first, then combine them at the end of each minibatch. If we assume that $\mathbf{A}_l$ and $\mathbf{B}_l$ are fixed,

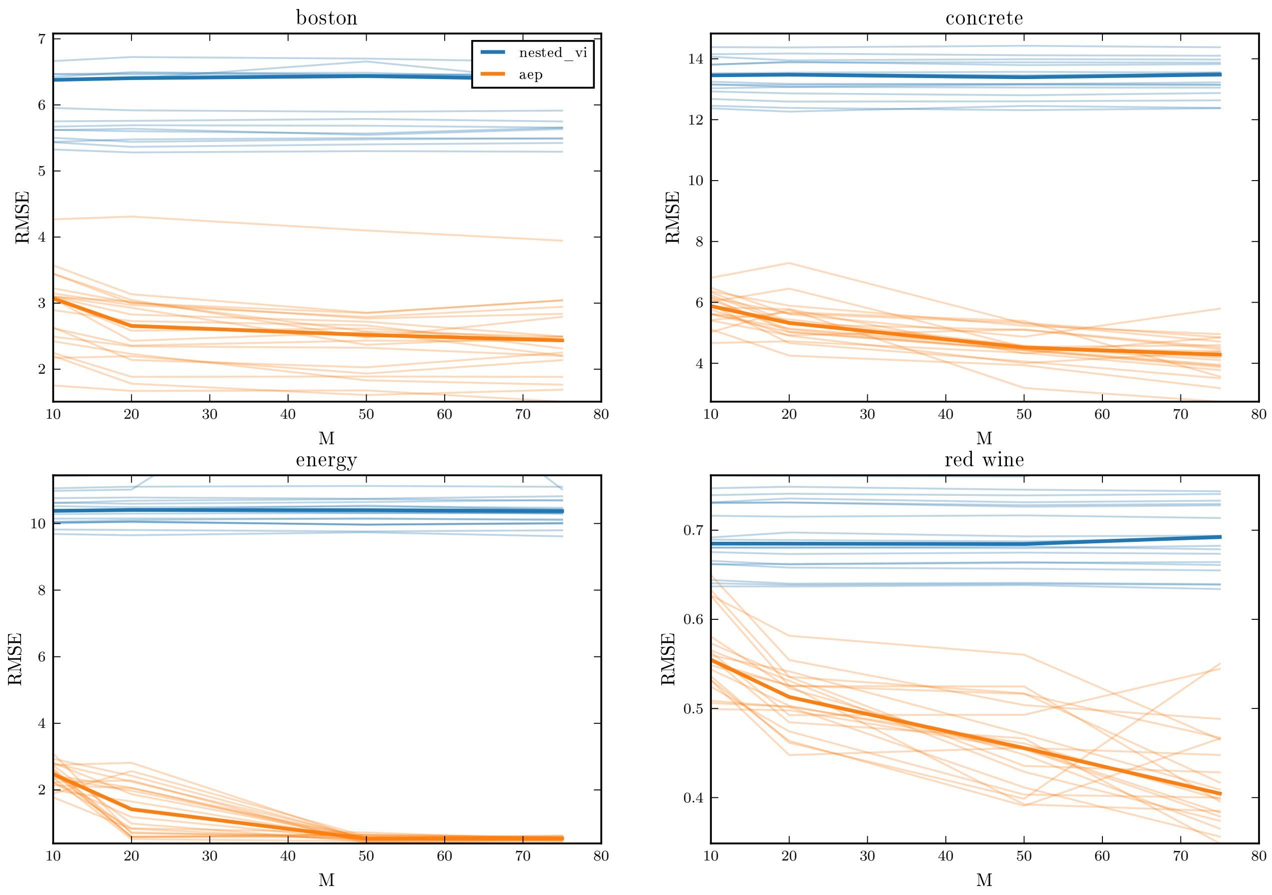

Figure 13. The test RMSE of the proposed approach and the Nested VI approach in (Hensman & Lawrence, 2014). The lower the better. AEP outperforms Nested VI by a large margin.

the gradients of m_l and v_l are as follows

$$\frac{\mathrm{d}m_l}{\mathrm{d}\beta_l} = \frac{\mathrm{d}\psi_{l,1}}{\mathrm{d}\beta_l}\mathbf{A}_l \tag{8}$$

$$\frac{\mathrm{d}v_l}{\mathrm{d}\beta_l} = \frac{\mathrm{d}\sigma_l^2}{\mathrm{d}\beta_l} + \frac{\mathrm{d}\psi_{l,0}}{\mathrm{d}\beta_l} + \mathrm{tr}\left(\mathbf{B}_l\frac{\mathrm{d}\psi_{l,2}}{\mathrm{d}\beta_l}\right) - 2m_l\frac{\mathrm{d}m_l}{\mathrm{d}\beta_l} \tag{9}$$

$$\frac{\mathrm{d}m_l}{\mathrm{d}\mathbf{A}_l} = \psi_{l,1}^{\intercal} \tag{10}$$

$$\frac{\mathrm{d}m_l}{\mathrm{d}\mathbf{B}_l} = \mathbf{0} \tag{11}$$

$$\frac{\mathrm{d}v_l}{\mathrm{d}\mathbf{A}_l} = -2m_l\frac{\mathrm{d}m_l}{\mathrm{d}\mathbf{A}_l} \tag{12}$$

$$\frac{\mathrm{d}v_l}{\mathrm{d}\mathbf{B}_l} = \psi_{l,2}^{\intercal} \tag{13}$$

At the end of the forward step, we can obtain $Z = q(y) = \mathcal{N}(y; m_L, v_L)$, leading to,

$$\log\mathcal{Z} = -\frac{1}{2}\log(2\pi v_L) - \frac{1}{2}\frac{(y-m_L)^2}{v_L} \tag{14}$$

$$\frac{\mathrm{d}\log\mathcal{Z}}{\mathrm{d}m_L} = \frac{y-m_L}{v_L} \tag{15}$$

$$\frac{\mathrm{d}\log\mathcal{Z}}{\mathrm{d}v_L} = -\frac{1}{2v_L} + \frac{1}{2}\frac{(y-m_L)^2}{v_L^2}. \tag{16}$$

We are now ready to perform the backpropagation step, that is we compute the gradients of $\log\mathcal{Z}$ w.r.t. parameters at a layer α_l using the chain rule,

$$\frac{\mathrm{d}\log\mathcal{Z}}{\mathrm{d}\alpha_l} = \frac{\mathrm{d}\log\mathcal{Z}}{\mathrm{d}m_l}\frac{\mathrm{d}m_l}{\mathrm{d}\alpha_l} + \frac{\mathrm{d}\log\mathcal{Z}}{\mathrm{d}v_l}\frac{\mathrm{d}v_l}{\mathrm{d}\alpha_l}. \tag{17}$$

Similarly, we can compute the gradients w.r.t. the mean and variance of the input distribution, m_{l-1} and v_{l-1}, and $\mathbf{A}_l$ and $\mathbf{B}_l$.

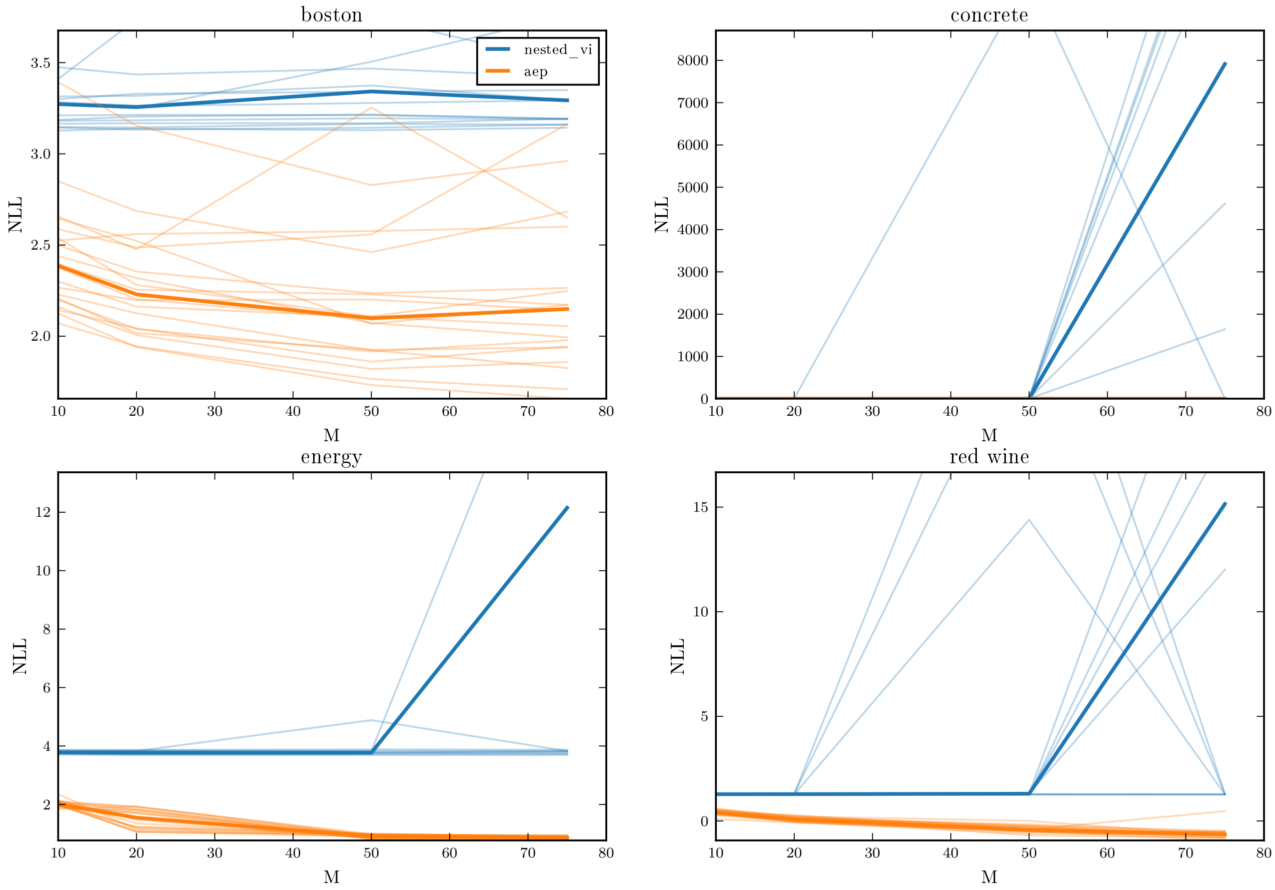

Figure 14. The test NLL of the proposed approach and the Nested VI approach in (Hensman & Lawrence, 2014). The lower the better. AEP outperforms Nested VI by a large margin.

6. Computing the gradients of the approximate marginal likelihood

The approximate marginal likelihood as discussed in the main text is as follows,

$$\mathcal{F} = -(N-1)\phi(\theta) + N\phi(\theta^{\backslash 1}) - \phi(\theta_{\text{prior}}) + \sum_{n=1}^{N} \log \mathcal{Z}_n \tag{18}$$

where $\theta, \theta^{\backslash 1}$ and θ_{prior} are the natural parameters of $q(\mathbf{u})$, $q^{\backslash 1}(\mathbf{u})$ and $p(\mathbf{u})$ respectively, $\phi(\theta)$ is the log normaliser or log partition function of a Gaussian distribution with natural parameters θ or mean $\mathbf{m}$ and covariance $\mathbf{V}$,

$$\phi(\theta) = \frac{1}{2}\log|\mathbf{V}| + \frac{1}{2}\mathbf{m}^{\intercal}\mathbf{V}^{-1}\mathbf{m}, \tag{19}$$

α is the model hyperameters that we need to tune, and $\log \mathcal{Z}_n = \log \int q^{\backslash n}(\mathbf{u}) p(y_n|\mathbf{u}, \mathbf{X}_n) \mathrm{d}\mathbf{u}$. Consider the gradient of this objective function w.r.t. one parameter α_i,

$$\begin{aligned}
\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\alpha_i} &= -(N-1)\frac{\mathrm{d}\phi(\theta)}{\mathrm{d}\alpha_i} + N\frac{\mathrm{d}\phi(\theta^{\backslash 1})}{\mathrm{d}\alpha_i} \\
&\quad - \frac{\mathrm{d}\phi(\theta_{\text{prior}})}{\mathrm{d}\alpha_i} + \sum_{n=1}^{N}\frac{\mathrm{d}\log\mathcal{Z}_n}{\mathrm{d}\alpha_i} \\
&= -(N-1)\frac{\mathrm{d}\phi(\theta)}{\mathrm{d}\theta}\frac{\mathrm{d}\theta}{\mathrm{d}\alpha_i} + N\frac{\mathrm{d}\phi(\theta^{\backslash 1})}{\mathrm{d}\theta^{\backslash 1}}\frac{\mathrm{d}\theta^{\backslash 1}}{\mathrm{d}\alpha_i} \\
&\quad - \frac{\mathrm{d}\phi(\theta_{\text{prior}})}{\mathrm{d}\theta_{\text{prior}}}\frac{\mathrm{d}\theta_{\text{prior}}}{\mathrm{d}\alpha_i} + \sum_{n=1}^{N}\frac{\mathrm{d}\log\mathcal{Z}_n}{\mathrm{d}\alpha_i} \\
&= -(N-1)\eta^{\intercal}\frac{\mathrm{d}\theta}{\mathrm{d}\alpha_i} + N\eta^{\backslash 1,\intercal}\frac{\mathrm{d}\theta^{\backslash 1}}{\mathrm{d}\alpha_i} \\
&\quad - \eta_{\text{prior}}^{\intercal}\frac{\mathrm{d}\theta_{\text{prior}}}{\mathrm{d}\alpha_i} + \sum_{n=1}^{N}\frac{\mathrm{d}\log\mathcal{Z}_n}{\mathrm{d}\alpha_i}
\end{aligned}$$

where η, $\eta_{\setminus 1}$ and η_{prior} are the expected sufficient statistics under the $q(\mathbf{u})$, $q^{\setminus 1}(\mathbf{u})$ and $p(\mathbf{u})$ respectively. Specifically, for Gaussian approximate EP as discussed in the main paper, the natural parameters are as follows,

$$\begin{aligned}
&q(\mathbf{u}) : \theta = \theta_{\text{prior}} + N\theta_g \\
&q^{\setminus 1}(\mathbf{u}) : \theta^{\setminus 1} = \theta_{\text{prior}} + (N-1)\theta_g \\
&p(\mathbf{u}) : \theta_{\text{prior}}
\end{aligned}$$

leading to

$$\begin{aligned}
\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\alpha_i} &= \left[-(N-1)\eta^{\intercal} + N\eta^{\setminus 1,\intercal} - \eta_{\text{prior}}^{\intercal}\right] \frac{\mathrm{d}\theta_{\text{prior}}}{\mathrm{d}\alpha_i} \\
&\quad + N(N-1)\left[-\eta^{\intercal} + \eta^{\setminus 1,\intercal}\right] \frac{\mathrm{d}\theta_g}{\mathrm{d}\alpha_i} + \sum_{n=1}^{N} \frac{\mathrm{d}\log \mathcal{Z}_n}{\mathrm{d}\alpha_i}
\end{aligned}$$

7. Dealing with non-Gaussian likelihoods

In this section, we discuss how to compute the log of $\mathcal{Z} = \int \mathrm{d}\mathbf{u}\, q^{\setminus 1}(\mathbf{u})\, p(y|\mathbf{u}, \mathbf{x})$ when we have a non-Gaussian likelihood $p(y|\mathbf{u}, \mathbf{x})$. For example, if the observations are binary, we can use the probit likelihood, that is $p(y|\mathrm{f}_L, h_{L-1}) = \phi(y\mathrm{f}_L)$ where ϕ is the Gaussian cdf. We now need to compute,

$$\begin{aligned}
\mathcal{Z} &= \int q^{\setminus 1}(\mathbf{u}) p(y|\mathbf{u}, \mathbf{x}) \mathrm{d}\mathbf{u} \\
&= \int q^{\setminus 1}(\mathbf{u}) p(\mathrm{f}_L | h_{L-1}, \mathbf{u}_L) p(y|\mathrm{f}_L) \mathrm{d}\mathbf{u} \mathrm{d}h_{L-1} \mathrm{d}\mathrm{f}_L \\
&\approx \int \mathcal{N}(\mathrm{f}_L; m_{\mathrm{f}}, v_{\mathrm{f}}) p(y|\mathrm{f}_L) \mathrm{d}\mathrm{f}_L
\end{aligned}$$

where we can find $q(\mathrm{f}_L) = \mathcal{N}(\mathrm{f}_L; m_{\mathrm{f}}, v_{\mathrm{f}})$ using the forward pass of the probabilistic backpropagation. The final integral above can be computed exactly, leading to,

$$\mathcal{Z} \approx \phi\left(\frac{y m_{\mathrm{f}}}{\sqrt{v_{\mathrm{f}} + 1}}\right)$$

If we have a different likelihood and there is no simple approximation available as above, we can evaluate $\mathcal{Z}$ by Monte Carlo averaging, that is to draw samples from $q(\mathrm{f}_L)$, evaluate the likelihood, then sum and normalise accordingly. However, as we are interested in $\log \mathcal{Z}$ and its gradients, the objective and gradients obtained by Monte Carlo will be slightly biased. This bias is, however, can be significantly reduced by using more samples.

8. Improving the Gaussian approximation

In this section, we discuss how to obtain a non-diagonal Gaussian approximation for the hidden variables from the second layer and above, when computing $\log \mathcal{Z}$. Consider a DGP with two GP layer, a one dimensional hidden layer and two dimensional observations $\mathbf{y} = [y_1, y_2]$. Following the derivation in the main text, we can exactly marginalise out the inducing outputs for each GP layer:

$$\mathcal{Z} = \int \mathrm{d}h_1 q(\mathbf{y}|h_1) q(h_1) \tag{20}$$

where $q(h_1) = \mathcal{N}(h_1; m_1, v_1)$ and

$$\begin{aligned}
q(\mathbf{y}|h_1) &= \mathcal{N}(\mathbf{y}|h_1; \mathbf{m}_{\mathbf{y}|h_1}, \mathbf{V}_{\mathbf{y}|h_1}) \\
&= \mathcal{N}\left(\mathbf{y}|h_1; \begin{bmatrix} m_{y_1|h_1} \\ m_{y_2|h_1} \end{bmatrix}, \begin{bmatrix} v_{y_1|h_1} & 0 \\ 0 & v_{y_2|h_1} \end{bmatrix}\right)
\end{aligned}$$

since we assume that there are two independent GPs in the second layer, and the distribution above is a conditional given the input to the second layer, h_1. Importantly, we need to integrate out h_1 in eqn. (20). As such, the resulting distribution over $\mathbf{y}$ become a complicated distribution in which y_1 and y_2 are strongly correlated. Consequently, any approximation that breaks this dependency could be poor. We aim to approximate this distribution by a non-diagonal Gaussian with the same moments, that is in words, the approximating Gaussian will have the mean being the expected mean, and the new covariance being the expected covariance plus the covariance of the mean,

$$\mathbf{m}_{\mathbf{y}} = \mathrm{E}_{q(h_1)}[\mathbf{m}_{\mathbf{y}|h_1}] \tag{21}$$

$$\mathbf{V}_{\mathbf{y}} = \mathrm{E}_{q(h_1)}[\mathbf{V}_{\mathbf{y}|h_1}] + \mathrm{covar}_{q(h_1)}[\mathbf{m}_{\mathbf{y}|h_1}] \tag{22}$$

Substitute the mean and covariance of the conditional $q(\mathbf{y}|h_1)$ into the above expressions gives us,

$$\mathbf{m}_{\mathbf{y}} = \begin{bmatrix} \mathrm{E}_{q(h_1)}[m_{y_1|h_1}] \\ \mathrm{E}_{q(h_1)}[m_{y_2|h_1}] \end{bmatrix} \tag{23}$$

and

$$\begin{aligned}
\mathbf{V}_{\mathbf{y}} = &\begin{bmatrix} \mathrm{E}_{q(h_1)}[v_{y_1|h_1}] & 0 \\ 0 & \mathrm{E}_{q(h_1)}[v_{y_2|h_1}] \end{bmatrix} \\
&+ \begin{bmatrix} \mathrm{E}_{q(h_1)}[m_{y_1|h_1}^2] & \mathrm{E}_{q(h_1)}[m_{y_1|h_1} m_{y_2|h_1}] \\ \mathrm{E}_{q(h_1)}[m_{y_1|h_1} m_{y_2|h_1}] & \mathrm{E}_{q(h_1)}[m_{y_2|h_1}^2] \end{bmatrix} \\
&- \mathbf{m}_{\mathbf{y}} \mathbf{m}_{\mathbf{y}}^{\intercal}
\end{aligned} \tag{24}$$

Note that the diagonal elements of $\mathbf{V}_{\mathbf{y}}$ are identical to the expression for the variance in the main text for the single dimensional case.

References

Demšar, Janez. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

Hensman, James and Lawrence, Neil D. Nested variational compression in deep Gaussian processes. *arXiv preprint arXiv:1412.1370, visited on 25/05/2016*, 2014.