

A DERIVATION OF THE ELBO

The variational inference Evidence Lower Bound is defined as

$$\mathcal{L}(\Omega, \Theta, \{\sigma_l^2\}_{l=1}^{L-1}) = \mathbb{E}_{q(\{\mathbf{F}^l, \mathbf{A}^l\}_{l=1}^L)} \left[\log \frac{p(\mathbf{y}, \{\mathbf{F}^l, \mathbf{A}^l\}_{l=1}^L)}{q(\{\mathbf{F}^l, \mathbf{A}^l\}_{l=1}^L)} \right],$$

where, using our model specification:

$$\begin{aligned} p(\mathbf{y}, \{\mathbf{F}^l, \mathbf{A}^l\}_{l=1}^L) &= \prod_{n=1}^N p(y_n | \mathbf{f}_{:,n}^L) \prod_{n=1}^N \prod_{l=1}^L \prod_{h=1}^{H_l} p(f_{h,n}^l | \mathbf{a}_h^l) p(\mathbf{a}_h^l), \\ q(\{\mathbf{F}^l, \mathbf{A}^l\}_{l=1}^L) &= \prod_{n=1}^N \prod_{l=1}^L \prod_{h=1}^{H_l} p(f_{h,n}^l | \mathbf{a}_h^l) q(\mathbf{a}_h^l). \end{aligned}$$

Using these expressions, the ELBO takes the following form:

$$\begin{aligned} \mathcal{L} &= \mathbb{E}_{q(\{\mathbf{F}^l, \mathbf{A}^l\}_{l=1}^L)} \left[\log \frac{p(\mathbf{y}, \{\mathbf{F}^l, \mathbf{A}^l\}_{l=1}^L)}{q(\{\mathbf{F}^l, \mathbf{A}^l\}_{l=1}^L)} \right] \\ &= \mathbb{E}_{q(\{\mathbf{F}^l, \mathbf{A}^l\}_{l=1}^L)} \left[\log \frac{\prod_{n=1}^N p(y_n | \mathbf{f}_{:,n}^L) \prod_{n=1}^N \prod_{l=1}^L \prod_{h=1}^{H_l} p(f_{h,n}^l | \mathbf{a}_h^l) p(\mathbf{a}_h^l)}{\prod_{n=1}^N \prod_{l=1}^L \prod_{h=1}^{H_l} p(f_{h,n}^l | \mathbf{a}_h^l) q(\mathbf{a}_h^l)} \right] \\ &= \mathbb{E}_{q(\{\mathbf{F}^l, \mathbf{A}^l\}_{l=1}^L)} \left[\log \frac{\prod_{n=1}^N p(y_n | \mathbf{f}_{:,n}^L) \prod_{l=1}^L \prod_{h=1}^{H_l} p(\mathbf{a}_h^l)}{\prod_{l=1}^L \prod_{h=1}^{H_l} q(\mathbf{a}_h^l)} \right]. \end{aligned}$$

The expectation can be split in two terms:

$$\mathcal{L} = \mathbb{E}_{q(\{\mathbf{F}^l, \mathbf{A}^l\}_{l=1}^L)} \left[\log \prod_{n=1}^N p(y_n | \mathbf{f}_{:,n}^L) \right] + \mathbb{E}_{q(\{\mathbf{F}^l, \mathbf{A}^l\}_{l=1}^L)} \left[\log \frac{\prod_{l=1}^L \prod_{h=1}^{H_l} p(\mathbf{a}_h^l)}{\prod_{l=1}^L \prod_{h=1}^{H_l} q(\mathbf{a}_h^l)} \right].$$

The logarithm in the first term does not depend on the regression coefficients $\{\mathbf{A}^l\}_{l=1}^L$ and neither on $\{\mathbf{F}^l\}_{l=1}^{L-1}$. On the other hand, the logarithm on the second term does not depend on $\{\mathbf{F}^l\}_{l=1}^L$. Thus,

$$\begin{aligned} \mathcal{L} &= \mathbb{E}_{q(\mathbf{F}^L)} \left[\log \prod_{n=1}^N p(y_n | \mathbf{f}_{:,n}^L) \right] + \mathbb{E}_{q(\{\mathbf{A}^l\}_{l=1}^L)} \left[\log \frac{\prod_{l=1}^L \prod_{h=1}^{H_l} p(\mathbf{a}_h^l)}{\prod_{l=1}^L \prod_{h=1}^{H_l} q(\mathbf{a}_h^l)} \right] \\ &= \sum_{n=1}^N \mathbb{E}_q [\log p(y_n | \mathbf{f}_{:,n}^L)] - \sum_{l=1}^L \sum_{h=1}^{H_l} \text{KL}(q(\mathbf{a}_h^l) | p(\mathbf{a}_h^l)). \end{aligned}$$

B DERIVATION OF THE MARGINALS

The variational distribution $q(\{\mathbf{f}^l\}_{l=1}^L)$ factorizes as the product of Gaussian distributions:

$$\begin{aligned} q(\{\mathbf{F}^l\}_{l=1}^L) &= \prod_{n=1}^N \prod_{l=1}^L \prod_{h=1}^{H_l} q(f_{h,n}^l | \mathbf{f}_{:,n}^{l-1}) \\ &= \prod_{n=1}^N \prod_{l=1}^L \prod_{h=1}^{H_l} \int p(f_{h,n}^l | \mathbf{f}_{:,n}^{l-1}, \mathbf{a}_h^l) q(\mathbf{a}_h^l) d\mathbf{a}_h^l \\ &= \prod_{n=1}^N \prod_{l=1}^L \prod_{h=1}^{H_l} \mathcal{N}(f_{h,n}^l | \hat{m}_{h,n}^l(\mathbf{f}_{:,n}^{l-1}), \hat{v}_{h,n}^l(\mathbf{f}_{:,n}^{l-1})). \end{aligned}$$

As a result, the n^{th} marginal of the final layer depends only on the n^{th} marginals of the other layers. That is,

$$q(f_n^L) = \int \prod_{l=1}^L \prod_{h=1}^{H_l} \mathcal{N}(f_{h,n}^l | \hat{m}_{h,n}^l(\mathbf{f}_{:,n}^{l-1}), \hat{v}_{h,n}^l(\mathbf{f}_{:,n}^{l-1})) d\mathbf{f}_{:,n}^1, \dots, d\mathbf{f}_{:,n}^{L-1}.$$

C EXPERIMENTAL SETTINGS

To speed-up computations in DVIP, at each layer l the generative function that defines the IP prior is shared across units. That is, the function $g_{\theta_h^l}(\cdot, \mathbf{z})$ is the same for every dimension h in that layer. As a consequence, the prior IP samples only need to be generated once per layer, as in [Ma et al. \(2019\)](#). In the BNN prior of DVIP and VIP, we tune the prior mean and variance of each weight and bias by maximizing the corresponding estimate of the marginal likelihood. As no regularizer is used for the prior parameters, the prior mean and variances are constrained to be the same in a layer of the BNN. This configuration avoids over-fitting and leads to improved results. The positive effect of this constraint is shown in [Appendix D](#). In DGP we consider 100 shared inducing points in each layer. We use ADAM ([Kingma and Ba, 2015](#)) as the optimization algorithm, and we set the learning rate to 10^{-3} , in DVIP, as in [Ma et al. \(2019\)](#). In DGP we use 10^{-2} as the learning rate, as in [Salimbeni and Deisenroth \(2017\)](#). Unless indicated otherwise, in DVIP and DGP we use the input dimension as the layer dimensionality, *i.e.* $H_l = D$, for $l = 1, \dots, L - 1$. In DGP the kernel employed is RBF with ARD ([Rasmussen and Williams, 2006](#)). The batch size is 100. All methods are trained for 150,000 iterations unless indicated otherwise. In VIP we do not employ the marginal likelihood regularizer described in [Ma et al. \(2019\)](#), since the authors of that paper told us that they did not use it in practice. Similarly, we do not regularize the estimation of the prior IP covariances in VIP nor DVIP. The inner dimensions of DVIP and DGP are fixed to the minimum between the number of attributes of the dataset and 30, as in [Salimbeni and Deisenroth \(2017\)](#). We use $\alpha = 0.5$ for VIP, as suggested in [Ma et al. \(2019\)](#). The reason why $\alpha = 0$ is used in the experiments on DVIP is that the use of alpha-divergences requires the optimization of the hyper-parameter α , which was against our *no hand-tuning* approach. Moreover, even if a fixed value of α could perform well on average, its use would also require to propagate more than one Monte Carlo sample to get a low biased estimate of objective function, which slows down training. For these reasons, we decided to keep the standard VI ELBO. Future work may consider using alpha-divergences for training DVIP. The source code can be accessed using the Github Repository [DeepVariationalImplicitProcesses](#).

D IMPACT OF THE CONSTRAINED PRIOR

[Figure 5](#) shows, on a toy problem, the obtained predictive distributions and learned prior samples of VIP, for $\alpha = 0$. This corresponds to a 1 layer particular case of DVIP. We consider two cases: (1) using a full unconstrained prior BNN, in which the mean and variance of each weight and bias can be independently tuned, (shown above) and (2), the considered constrained BNN in which prior means and variances are shared across layers (shown below). The second approach considerably reduces the number of parameters in the prior and we observe that it generates smoother prior functions. The predictive distribution is also smoother than when the prior is unconstrained. However, despite providing better results by avoiding over-fitting, there might be datasets where using the full unconstrained parameterization of the BNN leads to improved results. For example, in problems where a more flexible prior may be beneficial to obtain good generalization properties on un-seen data.

Consider the Boston and the Energy datasets. To highlight that prior over-fitting is a dataset-specific matter, [Table 4](#) shows the obtained results using an unconstrained prior and a constrained prior for VIP on the aforementioned datasets. As one may see, significantly over-fitting is taking place on Boston. More precisely, the training error improves a lot when using the unconstrained prior. By contrast, test error and other performance metrics deteriorate on the test set. In Energy, however, a more flexible prior results in better test RMSE and CRPS, but worse test log-likelihood.

Table 4: Results on Boston and Energy dataset using the constrained and unconstrained BNN prior for VIP with $\alpha = 0$.

Unconstrained	RMSE Train	RMSE Test	NLL Train	NLL Test	CRPS Train	CRPS Test
Boston	0.05 \pm 0.00	5.85 \pm 0.14	-1.21 \pm 0.19	5126.07 \pm 274.79	0.03 \pm 0.00	4.31 \pm 0.08
Energy	0.14 \pm 0.00	0.57 \pm 0.01	-0.51 \pm 0.01	6.52 \pm 0.42	0.079 \pm 0.00	0.36 \pm 0.01
Constrained	RMSE Train	RMSE Test	NLL Train	NLL Test	CRPS Train	CRPS Test
Boston	3.90 \pm 0.02	4.73 \pm 0.24	2.77 \pm 0.00	23.03 \pm 0.07	2.06 \pm 0.01	2.40 \pm 0.08
Energy	2.35 \pm 0.03	2.57 \pm 0.08	2.27 \pm 0.01	2.07 \pm 0.02	1.28 \pm 0.01	1.27 \pm 0.04

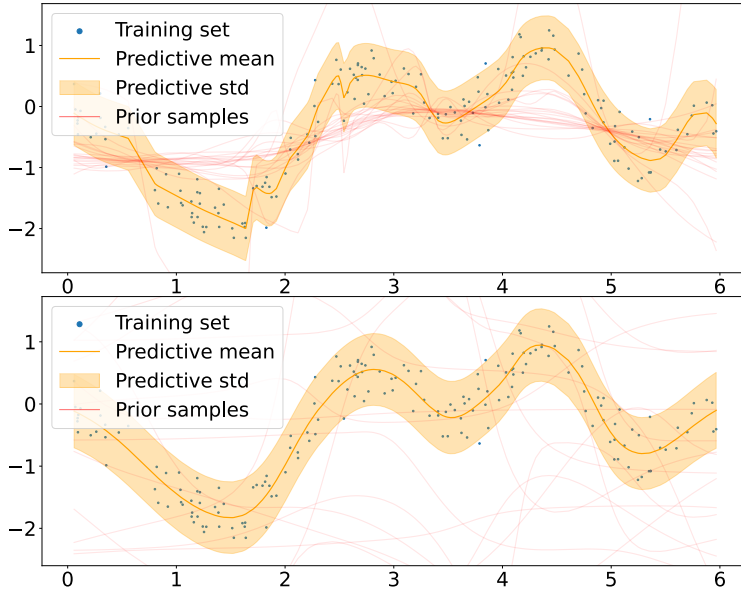


Figure 5: Resulting predictive distribution and prior samples over a toy dataset with full BNN prior (above) and constraint prior BNN (below).

E IMPACT OF THE NUMBER OF PRIOR SAMPLES

We explore the impact of the number of prior samples S has on the performance and training time of the proposed method. Table 5 shows the results obtained using DVIP with 3 layers on Protein and Power datasets (UCI). These results show how increasing the number of prior samples produces better results, in terms of performance, compared to using lower values of S . However, it is important to consider that this value scales quadratically the computational complexity of evaluating the ELBO, heavily influencing the training cost of the model.

Table 5: Results on Power and Protein datasets (UCI) using DVIP with 3 layers and different values of the number of prior samples S .

Power	$S = 10$	$S = 20$	$S = 30$	$S = 40$	$S = 50$
RMSE	4.03 ± 0.04	4.01 ± 0.04	3.94 ± 0.04	3.92 ± 0.04	3.94 ± 0.04
NLL	2.81 ± 0.01	2.81 ± 0.00	2.79 ± 0.01	2.78 ± 0.01	2.79 ± 0.01
CRPS	2.19 ± 0.01	2.18 ± 0.01	2.14 ± 0.01	2.11 ± 0.01	2.13 ± 0.01
CPU Time (s)	2693 ± 19	2806 ± 22	3152 ± 20	3451 ± 63	3742 ± 55
Protein	$S = 10$	$S = 20$	$S = 30$	$S = 40$	$S = 50$
RMSE	4.53 ± 0.01	4.40 ± 0.01	4.28 ± 0.01	4.27 ± 0.01	4.21 ± 0.01
NLL	2.92 ± 0.00	2.90 ± 0.00	2.87 ± 0.00	2.86 ± 0.00	2.85 ± 0.00
CRPS	2.52 ± 0.00	2.43 ± 0.00	2.36 ± 0.00	2.34 ± 0.00	2.31 ± 0.00
CPU Time (s)	2334 ± 28	2734 ± 19	3616 ± 11	3727 ± 35	4330 ± 52

F ROBUSTNESS OVER THE PRIOR BNN ARCHITECTURE

In this section, we study the impact that changing the prior BNN structure has over the performance of the proposed method. Table 6 shows the results obtained using DVIP with 2,3 and 4 layers on Protein and Power datasets (UCI) with two different BNN structures, 2 hidden layers with 20 units (20-20) and 3 hidden layers with 10 units (10-10-10). These results (that are to be compared with the original ones obtained using 2 hidden layers and 10 units on Table 11) show that changing the structure of the BNN does not heavily affect the obtained results, given that it is capable of learning similar function distributions with the different BNN architectures.

Table 6: Results on Power and Protein datasets (UCI) using DVIP with different prior BNN architectures.

	BNN 10-10-10			BNN 20-20		
Power	DVIP 2	DVIP 3	DVIP 4	DVIP 2	DVIP 3	DVIP 4
RMSE	4.02 ± 0.04	3.94 ± 0.04	3.97 ± 0.04	4.02 ± 0.04	3.99 ± 0.04	3.93 ± 0.04
NLL	2.81 ± 0.01	2.79 ± 0.00	2.80 ± 0.01	2.81 ± 0.01	2.80 ± 0.01	2.79 ± 0.01
CRPS	2.18 ± 0.01	2.13 ± 0.01	2.15 ± 0.01	2.18 ± 0.01	2.16 ± 0.01	2.13 ± 0.01
Protein	DVIP 2	DVIP 3	DVIP 4	DVIP 2	DVIP 3	DVIP 4
RMSE	4.57 ± 0.01	4.37 ± 0.01	4.29 ± 0.01	4.55 ± 0.01	4.43 ± 0.01	4.31 ± 0.01
NLL	2.93 ± 0.00	2.89 ± 0.00	2.87 ± 0.00	2.93 ± 0.00	2.90 ± 0.00	2.87 ± 0.00
CRPS	2.55 ± 0.00	2.42 ± 0.00	2.36 ± 0.00	2.54 ± 0.00	2.45 ± 0.00	2.37 ± 0.00

G USING A GP AS PRIOR IN DVIP AND VIP

In this section we investigate the use of a GP prior in DVIP to approximate GPs and hence DGPs. From a theoretical point of view, GP samples could be used in VIPs prior, ensuring that VIP does converge to a GP when the number of prior samples (and linear regression coefficients) is large enough. As a result, DVIP can converge to a DGP. However, DVIP needs to evaluate covariances among the process values at the training points. This requires taking continuous samples from a GP, something that is not possible in practice unless one samples the process values at all the training points, something that is intractable for big datasets. To surpass this limitation, a BNN with a single layer of cosine activation functions can be used to approximate a GP with RBF kernel (Rahimi and Recht, 2007). Generating continuous samples from this BNN is easy. One only has to sample the weights from their corresponding prior distribution. However, in order to correctly approximate the GP, a large number of hidden units is needed in the BNN, increasing the computational cost of taking such samples.

Furthermore, in many situations the predictive distribution of a sparse GP is very different of that of a full GP. Meaning that even when using an approximate GP prior in VIP, by means of a BNN with enough hidden units, it may not be enough to accurately approximate a sparse GP. Specifically, we have observed that this is the case in the Year dataset. In this dataset, the difference in performance between DVIP and DGP is not only a consequence of the different prior, but also the posterior approximation. More precisely, DVIP uses a linear regression approximation to the GP, while a DGP uses an inducing points based approximation. To show this, we have also implemented an inducing points approximation on VIP. For this, the required covariance matrices are estimated using a large number of prior samples. The obtained results can be seen in Table 7. There, we show the results of VIP using an approximated GP prior, using both the linear regression approximation and an approximation based on inducing points. We also report the results of the sparse GP and the average training time of each method in seconds. The table shows that VIP can be used with inducing points to approximate a GP. Specifically, the inducing points approximation of VIP gives very similar results to those of a sparse GP. However, this is achieved at a prohibitive training time. The computational bottlenecks are the GP approximation using a wide single layer BNN (we used 2000 units in the hidden layer), and the generation of a large number of prior samples from the BNN to approximate the covariance matrix (we used 2000 samples). Given the high computational cost of training a VIP model on this dataset, considering DVIP with more than 1 layer is too expensive.

Table 7: Results on Year dataset of VIP with the usual BNN prior with 2 hidden units of width 10 and tanh activations, VIP using a BNN that approximates a GP with RBF kernel, VIP with the last prior and 100 inducing points and a sparse GP with 100 inducing points. Experiments with VIP are trained using $\alpha = 0$.

Year	VIP	VIP-GP (linear regression)	VIP-GP (inducing points)	SGP
RMSE	10.27 ± 0.01	10.23 ± 0.01	9.28 ± 0.01	9.15 ± 0.01
NLL	3.74 ± 0.00	3.77 ± 0.00	3.64 ± 0.00	3.62 ± 0.00
CRPS	5.45 ± 0.01	5.45 ± 0.02	4.85 ± 0.01	4.83 ± 0.01
CPU Time (s)	1217 ± 257	1687 ± 271	30867 ± 326	1874 ± 265

We have carried out extra experiments in the smaller datasets Protein and Kin8nm to assess if using a GP prior in the context of DVIP generates results that are closer to those of a DGP. These are the regression datasets from the UCI repository where DGPs performed better than DVIP. In this case, we did not consider the inducing points approximation of the GP, as in the previous paragraph, but the linear regression approximation of VIP. We include results for (i) DVIP using the initially proposed BNN prior with 2 layers of 10 hidden units and tanh activation functions; (ii) DVIP using the wide BNN that approximates the prior GP; (iii) DGPs using sparse GPs in each layer based on inducing points. The results obtained are shown in Tables 8 and 9. We observe that in both cases using a GP prior in DVIP often improves results and performs similarly to a DGP, especially for a large number of layers L , even when there are differences in the approximation of the posterior distribution, *i.e.*, DVIP uses a linear regression approximation to the GP and DGP uses inducing points. Again, using a wide BNN to approximate the prior GP results in a significant increment of the training time, making DVIP slower than DGP.

Table 8: Results on Protein UCI dataset using DVIP with the usual BNN prior, the approximated GP prior and deep GPs. Experiments with VIP are trained using $\alpha = 0$.

BNN Prior	VIP	DVIP 2	DVIP 3	DVIP 4	DVIP 5
RMSE	4.76 \pm 0.01	4.24 \pm 0.01	4.14 \pm 0.01	4.14 \pm 0.01	4.09 \pm 0.01
NLL	2.98 \pm 0.00	2.86 \pm 0.00	2.84 \pm 0.00	2.84 \pm 0.00	2.83 \pm 0.00
CRPS	2.68 \pm 0.00	2.34 \pm 0.00	2.26 \pm 0.00	2.25 \pm 0.00	2.21 \pm 0.00
CPU time (s)	3086 \pm 173	3981 \pm 182	8604 \pm 774	9931 \pm 616	12568 \pm 327
GP Prior	VIP	DVIP 2	DVIP 3	DVIP 4	DVIP 5
RMSE	4.89 \pm 0.01	4.26 \pm 0.01	4.07 \pm 0.01	4.02 \pm 0.01	4.01 \pm 0.01
NLL	3.00 \pm 0.00	2.87 \pm 0.00	2.82 \pm 0.00	2.81 \pm 0.00	2.81 \pm 0.00
CRPS	2.77 \pm 0.00	2.35 \pm 0.00	2.21 \pm 0.00	2.17 \pm 0.00	2.16 \pm 0.00
CPU time (s)	5880 \pm 249	12293 \pm 564	20351 \pm 1110	18514 \pm 575	25835 \pm 1259
DS-DGP	SGP	DGP 2	DGP 3	DGP 4	DGP 5
RMSE	4.56 \pm 0.01	4.17 \pm 0.01	4.00 \pm 0.01	4.01 \pm 0.01	4.02 \pm 0.01
NLL	2.93 \pm 0.00	2.84 \pm 0.00	2.79 \pm 0.00	2.79 \pm 0.00	2.80 \pm 0.00
CRPS	2.56 \pm 0.00	2.31 \pm 0.00	2.19 \pm 0.00	2.19 \pm 0.00	2.20 \pm 0.00
CPU time (s)	2690 \pm 114	10031 \pm 129	17528 \pm 689	16128 \pm 190	20653 \pm 969

Table 9: Results on Kin8nm UCI dataset using DVIP with the usual BNN prior, the approximated GP prior and deep GPs. Experiments with VIP are trained using $\alpha = 0$.

BNN Prior	VIP	DVIP 2	DVIP 3	DVIP 4	DVIP 5
RMSE	0.15 \pm 0.00	0.07 \pm 0.00	0.07 \pm 0.00	0.07 \pm 0.00	0.07 \pm 0.00
NLL	-0.47 \pm 0.00	-1.13 \pm 0.00	-1.18 \pm 0.00	-1.16 \pm 0.00	-1.17 \pm 0.00
CRPS	0.08 \pm 0.00	0.04 \pm 0.00	0.04 \pm 0.00	0.04 \pm 0.00	0.04 \pm 0.00
CPU time (s)	2109 \pm 57	5086 \pm 232	6927 \pm 27	9459 \pm 77	11763 \pm 141
GP Prior	VIP	DVIP 2	DVIP 3	DVIP 4	DVIP 5
RMSE	0.14 \pm 0.00	0.07 \pm 0.00	0.06 \pm 0.00	0.06 \pm 0.00	0.06 \pm 0.00
NLL	-0.43 \pm 0.00	-1.20 \pm 0.00	-1.25 \pm 0.00	-1.26 \pm 0.00	-1.25 \pm 0.00
CRPS	0.08 \pm 0.00	0.04 \pm 0.00	0.03 \pm 0.00	0.03 \pm 0.00	0.03 \pm 0.00
CPU time (s)	6672 \pm 442	14300 \pm 847	17573 \pm 1033	22561 \pm 980	22669 \pm 657
DS-DGP	SGP	DGP 2	DGP 3	DGP 4	DGP 5
RMSE	0.09 \pm 0.00	0.06 \pm 0.00	0.06 \pm 0.00	0.06 \pm 0.00	0.06 \pm 0.00
NLL	-0.91 \pm 0.00	-1.29 \pm 0.00	-1.32 \pm 0.00	-1.33 \pm 0.00	-1.30 \pm 0.00
CRPS	0.05 \pm 0.00	0.03 \pm 0.00	0.03 \pm 0.00	0.03 \pm 0.00	0.03 \pm 0.00
CPU time (s)	2053 \pm 81	6375 \pm 331	11147 \pm 472	17502 \pm 1060	21846 \pm 1246

As a conclusion, DVIPs general and flexible definition allows the usage of (approximate) GP priors and inducing points. This enables performing (nearly) equally to a sparse GP or a DGP. However, the computational overhead of doing these approximations is prohibitive. Thus, if a GP prior is to be considered, it is a much better approach to just use the DGP framework, and leave DVIP to cases where flexible but easy-to-sample-from priors can be used.

H FURTHER RESULTS

The results regarding the **regression UCI benchmark datasets** are provided in Table 11. In addition, results on **large scale regression datasets** are given in Table 10.

Table 10: Negative Log Likelihood and Continuous Ranked Probability Score results on **large scale regression datasets**.

NLL	Single-layer		Ours				DS-DGP
	SGP	VIP	DVIP 2	DVIP 3	DVIP 4	DVIP 5	Best DGP
Year	3.62 ± 0.00	3.74 ± 0.00	3.68 ± 0.00	3.64 ± 0.00	3.64 ± 0.00	3.63 ± 0.00	3.59 ± 0.00
Airline	5.10 ± 0.00	5.11 ± 0.00	5.08 ± 0.00	5.07 ± 0.00	5.07 ± 0.00	5.06 ± 0.00	5.07 ± 0.00
Taxi	7.73 ± 0.00	7.73 ± 0.00	7.72 ± 0.00	7.69 ± 0.00	7.72 ± 0.00	7.70 ± 0.00	7.73 ± 0.00
CRPS	Single-layer		Ours				DS-DGP
	SGP	VIP	DVIP 2	DVIP 3	DVIP 4	DVIP 5	Best DGP
Year	4.83 ± 0.01	5.45 ± 0.01	5.13 ± 0.02	4.96 ± 0.01	4.93 ± 0.01	4.91 ± 0.02	4.680 ± 0.01
Airline	17.90 ± 0.05	17.93 ± 0.04	17.53 ± 0.05	17.54 ± 0.07	17.51 ± 0.05	17.45 ± 0.04	17.47 ± 0.03
Taxi	283.79 ± 0.19	284.22 ± 0.20	282.09 ± 0.32	274.65 ± 0.68	281.28 ± 0.44	277.60 ± 0.90	282.99 ± 0.21

In order to complement the **missing values interpolation** results on the CO₂ dataset. The same experiment is repeated on a Deep GP with a single layer, that is, an sparse GP. The configuration that has been used for all of the experiments is kept here. These results show that a single sparse GP does also suffer from the same problem that Deep GPs. That is, the mean reversion problem when facing a (wide enough) gap in the training dataset.

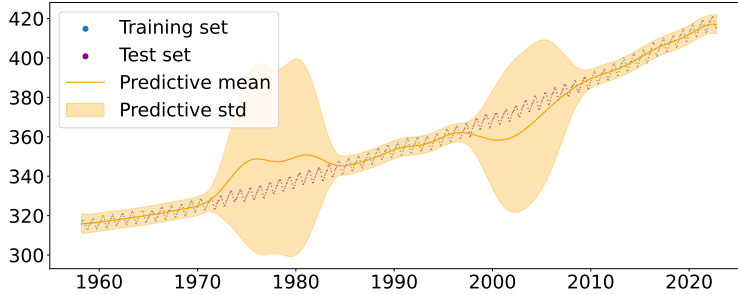


Figure 6: **Missing values interpolation** results on the CO₂ dataset. Predictive distribution of a sparse GP (Deep GP with a single layer). Two times the standard deviation is represented.

I POTENTIAL NEGATIVE SOCIETAL IMPACTS

Given that machine learning models are increasingly being used to make decisions that have a significant impact on society, industry, and individuals (e.g. autonomous vehicle safety [McAllister et al. \(2017\)](#), disease detection [Sajda \(2006\)](#); [Singh \(2021\)](#)), it is critical that we have a thorough understanding of the methods used and can provide rigorous performance guarantees. We conscientiously studied the performance and application of DVIP to different kinds of datasets and tasks as part of our empirical evaluation, demonstrating its ability to adjust to each domain-specific dataset.

Table 11: Negative Log Likelihood, Root Mean Squared Error and Continuous Ranked Probability Score results on regression UCI benchmark datasets.

NLL	Single-layer				Ours				DS-DGP			
	SGP	VIP	VIP 200	SIP	DVIP 2	DVIP 3	DVIP 4	DVIP 5	DGP 2	DGP 3	DGP 4	DGP 5
Boston	2.62 ± 0.05	2.76 ± 0.05	2.69 ± 0.03	2.72 ± 0.03	2.85 ± 0.09	2.59 ± 0.06	2.67 ± 0.09	2.66 ± 0.08	2.63 ± 0.05	2.63 ± 0.05	2.64 ± 0.05	2.65 ± 0.05
Energy	1.54 ± 0.02	2.07 ± 0.02	2.07 ± 0.02	1.17 ± 0.02	0.76 ± 0.02	0.70 ± 0.01	0.70 ± 0.01	0.73 ± 0.01	0.72 ± 0.01	0.74 ± 0.01	0.72 ± 0.01	0.73 ± 0.01
Concrete	3.16 ± 0.01	3.45 ± 0.02	3.48 ± 0.01	3.60 ± 0.05	3.24 ± 0.04	3.20 ± 0.05	3.03 ± 0.02	3.06 ± 0.02	3.17 ± 0.01	3.20 ± 0.01	3.13 ± 0.01	3.12 ± 0.01
Winered	0.93 ± 0.01	0.94 ± 0.01	0.96 ± 0.01	0.97 ± 0.01	0.94 ± 0.01	0.94 ± 0.01	0.94 ± 0.01	0.95 ± 0.01	0.94 ± 0.01	0.94 ± 0.01	0.94 ± 0.01	0.93 ± 0.01
Power	2.84 ± 0.00	2.85 ± 0.00	2.86 ± 0.00	2.84 ± 0.00	2.82 ± 0.01	2.81 ± 0.00	2.79 ± 0.01	2.79 ± 0.01	2.81 ± 0.01	2.80 ± 0.00	2.80 ± 0.00	2.80 ± 0.01
Protein	2.93 ± 0.00	3.03 ± 0.00	3.03 ± 0.00	3.00 ± 0.00	2.93 ± 0.00	2.89 ± 0.00	2.88 ± 0.00	2.86 ± 0.00	2.84 ± 0.00	2.79 ± 0.00	2.79 ± 0.00	2.80 ± 0.00
Naval	-6.11 ± 0.06	-4.50 ± 0.02	-4.31 ± 0.00	-2.79 ± 0.00	-5.89 ± 0.02	-5.98 ± 0.01	-5.90 ± 0.01	-5.92 ± 0.01	-6.35 ± 0.09	-6.21 ± 0.04	-6.27 ± 0.06	-6.21 ± 0.08
Kin8nm	-0.91 ± 0.00	-0.31 ± 0.00	-0.25 ± 0.00	-0.27 ± 0.02	-1.00 ± 0.00	-1.13 ± 0.00	-1.15 ± 0.00	-1.16 ± 0.00	-1.29 ± 0.00	-1.32 ± 0.00	-1.33 ± 0.00	1.30 ± 0.00
RMSE	Single-layer				Ours				DS-DGP			
	SGP	VIP	VIP 200	SIP	DVIP 2	DVIP 3	DVIP 4	DVIP 5	DGP 2	DGP 3	DGP 4	DGP 5
Boston	3.48 ± 0.17	4.78 ± 0.28	4.49 ± 0.28	5.10 ± 0.32	3.87 ± 0.19	3.50 ± 0.20	3.60 ± 0.19	3.66 ± 0.21	3.51 ± 0.18	3.53 ± 0.19	3.55 ± 0.20	3.56 ± 0.20
Energy	1.07 ± 0.03	2.57 ± 0.08	2.68 ± 0.07	3.27 ± 0.09	0.52 ± 0.01	0.47 ± 0.01	0.46 ± 0.01	0.47 ± 0.01	0.46 ± 0.01	0.47 ± 0.01	0.46 ± 0.01	0.46 ± 0.01
Concrete	5.84 ± 0.12	7.75 ± 0.15	8.06 ± 0.16	8.70 ± 0.43	6.01 ± 0.16	5.68 ± 0.18	5.13 ± 0.12	5.27 ± 0.13	5.86 ± 0.12	6.01 ± 0.12	5.54 ± 0.11	5.52 ± 0.12
Winered	0.61 ± 0.00	0.62 ± 0.00	0.63 ± 0.00	0.64 ± 0.00	0.62 ± 0.00	0.62 ± 0.00	0.62 ± 0.00	0.62 ± 0.00	0.62 ± 0.00	0.62 ± 0.00	0.62 ± 0.00	0.62 ± 0.00
Power	4.15 ± 0.03	4.21 ± 0.03	4.22 ± 0.03	4.14 ± 0.03	4.06 ± 0.04	4.01 ± 0.04	3.97 ± 0.04	3.95 ± 0.04	4.00 ± 0.04	3.98 ± 0.03	3.99 ± 0.03	3.96 ± 0.04
Protein	4.56 ± 0.01	5.05 ± 0.01	5.04 ± 0.01	4.92 ± 0.02	4.54 ± 0.01	4.40 ± 0.01	4.33 ± 0.01	4.26 ± 0.01	4.17 ± 0.01	4.00 ± 0.01	4.01 ± 0.01	4.02 ± 0.01
Naval	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Kin8nm	0.09 ± 0.00	0.17 ± 0.00	0.18 ± 0.00	0.18 ± 0.00	0.08 ± 0.00	0.07 ± 0.00	0.07 ± 0.00	0.07 ± 0.00	0.06 ± 0.00	0.06 ± 0.00	0.06 ± 0.00	0.06 ± 0.00
CRPS	Single-layer				Ours				DS-DGP			
	SGP	VIP	VIP 200	SIP	DVIP 2	DVIP 3	DVIP 4	DVIP 5	DGP 2	DGP 3	DGP 4	DGP 5
Boston	1.79 ± 0.05	2.25 ± 0.08	2.13 ± 0.08	2.35 ± 0.11	1.91 ± .06	1.76 ± 0.07	1.81 ± 0.07	1.78 ± 0.06	1.79 ± 0.05	1.80 ± 0.06	1.80 ± 0.06	1.81 ± 0.06
Energy	0.62 ± 0.01	1.27 ± 0.04	1.30 ± 0.03	1.21 ± 0.04	0.28 ± 0.00	0.26 ± 0.00	0.26 ± 0.00	0.26 ± 0.00	0.26 ± 0.00	0.26 ± 0.00	0.26 ± 0.00	0.26 ± 0.00
Concrete	3.20 ± 0.05	4.29 ± 0.08	4.43 ± 0.08	4.69 ± 0.11	3.26 ± 0.07	3.03 ± 0.09	2.74 ± 0.05	2.83 ± 0.05	3.21 ± 0.05	3.31 ± 0.05	3.05 ± 0.05	3.04 ± 0.05
Winered	0.34 ± 0.00	0.34 ± 0.00	0.35 ± 0.00	0.35 ± 0.00	0.34 ± 0.00	0.34 ± 0.00	0.34 ± 0.00	0.34 ± 0.00	0.34 ± 0.00	0.34 ± 0.00	0.34 ± 0.00	0.34 ± 0.00
Power	2.27 ± 0.01	2.31 ± 0.01	2.31 ± 0.01	2.27 ± 0.01	2.21 ± 0.01	2.18 ± 0.01	2.14 ± 0.01	2.14 ± 0.01	2.17 ± 0.01	2.16 ± 0.01	2.17 ± 0.01	2.15 ± 0.01
Protein	2.56 ± 0.00	2.87 ± 0.00	2.86 ± 0.01	2.77 ± 0.00	2.54 ± 0.00	2.43 ± 0.00	2.38 ± 0.00	2.33 ± 0.00	2.31 ± 0.00	2.19 ± 0.00	2.19 ± 0.00	2.20 ± 0.00
Naval	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Kin8nm	0.05 ± 0.00	0.09 ± 0.0	0.10 ± 0.00	0.10 ± 0.00	0.04 ± 0.00	0.04 ± 0.00	0.04 ± 0.00	0.04 ± 0.00	0.03 ± 0.00	0.03 ± 0.00	0.03 ± 0.00	0.03 ± 0.00