Supplementary material

A. Hyperparameters and training settings

For all our models we used the following settings in the UCI experiments:

Kernels. All kernels were the RBF, using a lengthscale parameter per input dimension, initialized to the squareroot of the dimension.

Inducing points. For data with more than 128 training data points the inducing point locations were chosen using the `kmeans2` from the `scipy` package, with 128 points. Otherwise they were set to the data. For latent variable layers, the GP layer above has extra dimensions. The inducing points for the extra dimensions were padded with random draws from a standard normal variable.

Linear projections between layers. We implemented the linear mean functions and multioutput structure using a linear projection of 5 independent GPs concatenated with their inputs. We initialized the projection matrix to the first 5 principle components of the data concatenated with the identity matrix.

Amortization of the variational parameters. We used three layer fully connected network with skip connections between each layer. We used the $\tanh$ non-linearity with 10 units for the inner layers. We used the weight initialization from (Glorot and Bengio, 2010) and the exponential function to enforce positivity of the standard deviation parameters. We added a bias of -5 in the final layer to ensure the standard deviations were small at the start of optimization.

Likelihood. The likelihood variance was intialized 0.01.

Parameterizations. All positive model parameters were constrained to be positive using the softplus function, clipped to 10^{-6}. The variational parameters for the sparse GP layers were parameterized by the mean and the square root of the covariance.

Optimization. The final GP layers were optimized using natural gradients on the natural parameters, with an initial step size of 0.01. All other parameters were optimized using the Adam optimizer (Kingma and Ba, 2015) with all parameters set to their tensorflow default values except the initial step size of 0.005. Optimizing just final layer using natural gradients and the inner layers with the Adam optimizer in this way is show by Hebbal et al. (2019) to be an effective strategy in practice. We used a batch size of 512 and trained for 100K iterations, annealing the learning rate of both the adam and natural gradient steps by a factor of 0.98 per 1000 iterations. The mean functions and kernel projection matrices (the P matrices that correlate the outputs) were *not* optimized.

Importance weights. We used 5 importance weights for all the latent variables models with importance-weighted variational inference.

Predictions. We used 2000 samples for prediction, sampling from the prior for w for the latent variables models. We then used a kernel density estimator (with Silverman's rule to set the bandwidth) to estimate the density.

Splits and preprocessing. We used 90% splits on the shuffled data, and rescaled the inputs and outputs to unit standard deviation and zero mean. For dataset with more than 10K test points we used a random sample of 10K test points (the same sample for each split). NB we have reported the test log-likelihoods without restoring the scaling, to facilitate comparisons between data. The splits were the same for all the models. We implemented our models using gpflow (Matthews et al., 2017).

Comparison of methods. The error bars in Table 1 are standard errors over the five splits. These error bars are likely to *overestimate* the uncertainty for comparisons between methods, however, as there may be correlations between the performance of methods over the splits (as the splits are the same for each method). The average ranks were computed for each split separately, and then averaged over the splits and method. To mitigate the effect of correlations between splits we report the median difference in test log-likelihood from the single layer GP.

VAE models The conditional VAE models used $\tanh$ activations and weight initializations from (Glorot and Bengio, 2010). Optimization was performed used the adam optimizer, with the same batch size and learning rate schedule as the GP models.

B. Variational posterior derivation

Here we derive the sparse GP posterior $q(f)$ and its KL divergence from the prior, following Hensman et al. (2013). This derivation applies to all the GP layers in the DGP model.

We begin with the approach of Titsias (2009) and form a variational distribution over the function f by conditioning the prior on a set of inducing points $\{f(\tilde{x}_i)\}_{i=1}^M$. We write $\tilde{\mathbf{f}}$ for the vector with ith component $f(\tilde{x}_i)$. In Titsias (2009), $q(f)$ is defined as

$$q(f) = p(f|\tilde{\mathbf{f}})q(\tilde{\mathbf{f}})\,, \tag{12}$$

where $q(\tilde{\mathbf{f}})$ is unspecified. In the single layer case, analytic results can be used to show that $q(\tilde{\mathbf{f}})$ has an optimal form that is Gaussian. Instead, we follow Hensman et al. (2013) and *assume* that

$$q(\tilde{\mathbf{f}}) = \mathcal{N}(\tilde{\mathbf{f}}|\tilde{\mathbf{m}}, \tilde{\mathbf{S}})\,. \tag{13}$$

As this choice is conjugate to the prior conditional $p(f|\tilde{\mathbf{f}})$, we can use standard Gaussians identities to show that

$$q(f) = \mathcal{GP}(\mu, \Sigma)\,, \tag{14}$$

where

$$\mu(x) = \mathbf{k}(x)^\top \tilde{\mathbf{K}}^{-1}\mathbf{m} \tag{15}$$

$$\Sigma(x, x') = k(x, x') - \mathbf{k}(x)^\top \tilde{\mathbf{K}}^{-1}\left(\tilde{\mathbf{K}} - \mathbf{S}\right)\tilde{\mathbf{K}}^{-1}\mathbf{k}(x) \tag{16}$$

with $[\mathbf{k}(x)]_i = k(x, \tilde{x}_i)$ and $[\tilde{\mathbf{K}}]_{ij} = k(\tilde{x}_i, \tilde{x}_j)$ The KL divergence is given by

$$\mathrm{KL}(q(f)||p(f)) = -\mathbb{E}_{q(f)}\log\frac{p(f)}{q(f)} \tag{17}$$

$$= -\mathbb{E}_{q(f)}\log\frac{p(f|\tilde{\mathbf{f}})p(\tilde{\mathbf{f}})}{p(f|\tilde{\mathbf{f}})q(\tilde{\mathbf{f}})} \tag{18}$$

$$= -\mathbb{E}_{q(\tilde{\mathbf{f}})}\log\frac{p(\tilde{\mathbf{f}})}{q(\tilde{\mathbf{f}})} \tag{19}$$

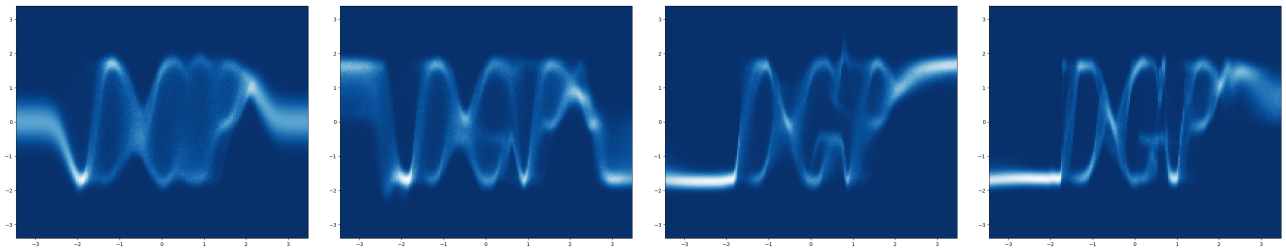
$$= \mathrm{KL}(q(\tilde{\mathbf{f}})||p(\tilde{\mathbf{f}}))\,, \tag{20}$$

which is finite.

The parameters $\tilde{\mathbf{m}}$ and $\tilde{\mathbf{S}}$ together with $\{\tilde{x}_i\}$ are variational parameters.

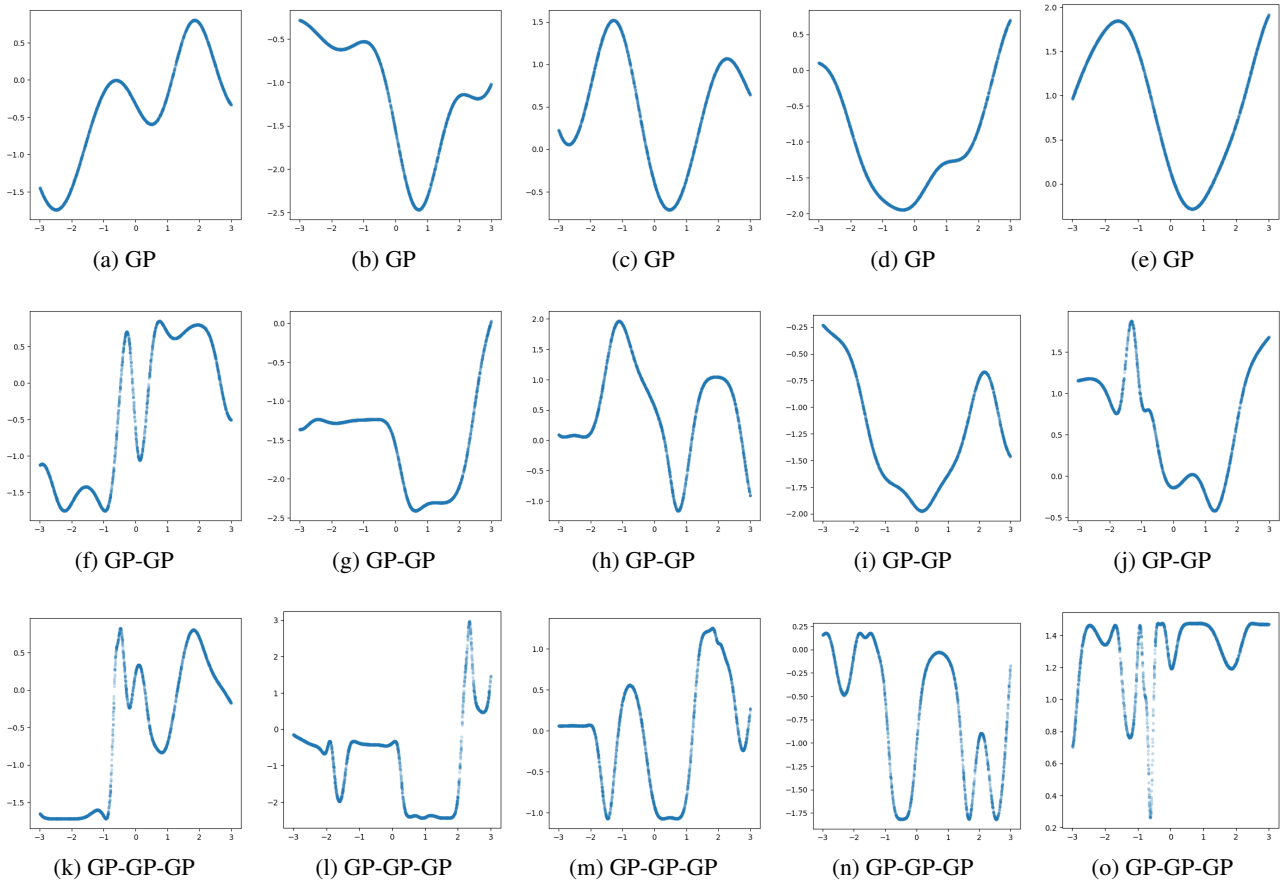
For multiple outputs, use the same matrix P for correlating the outputs as the prior. In practice, we implement the correlated output model by multiplying independent GPs by the matrix P, so we use independent outputs for the variational distribution.

C. Further tables and figures



(a) LV-GP, VI (b) LV-GP-GP, VI (c) LV-GP-GP-GP, VI (d) LV-GP-GP-GP, IW

Figure 5: Posteriors for further models for the '*DGP*' data

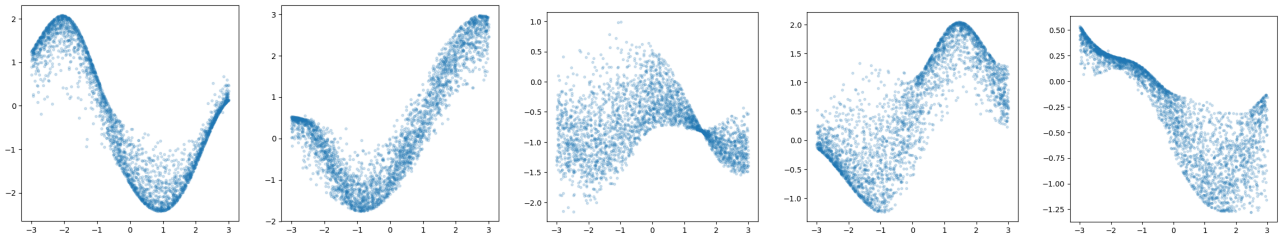


(a) GP (b) GP (c) GP (d) GP (e) GP

(f) GP-GP (g) GP-GP (h) GP-GP (i) GP-GP (j) GP-GP

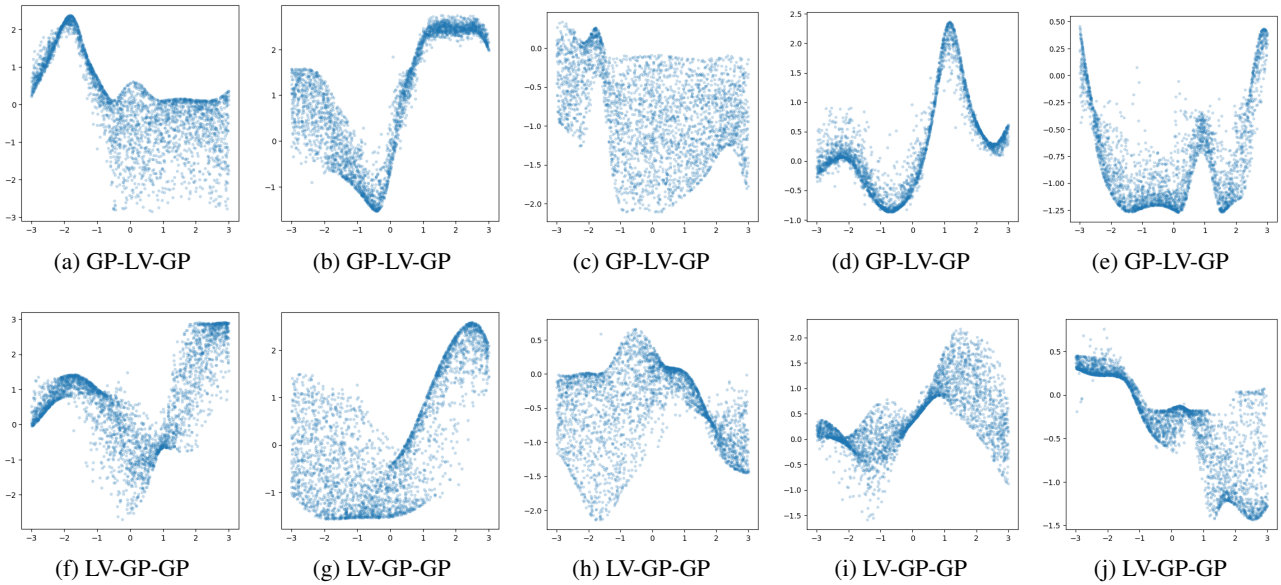
(k) GP-GP-GP (l) GP-GP-GP (m) GP-GP-GP (n) GP-GP-GP (o) GP-GP-GP

Figure 6: Prior draws from one, two and three layer models without latent variables



(a) LV-GP (b) LV-GP (c) LV-GP (d) LV-GP (e) LV-GP

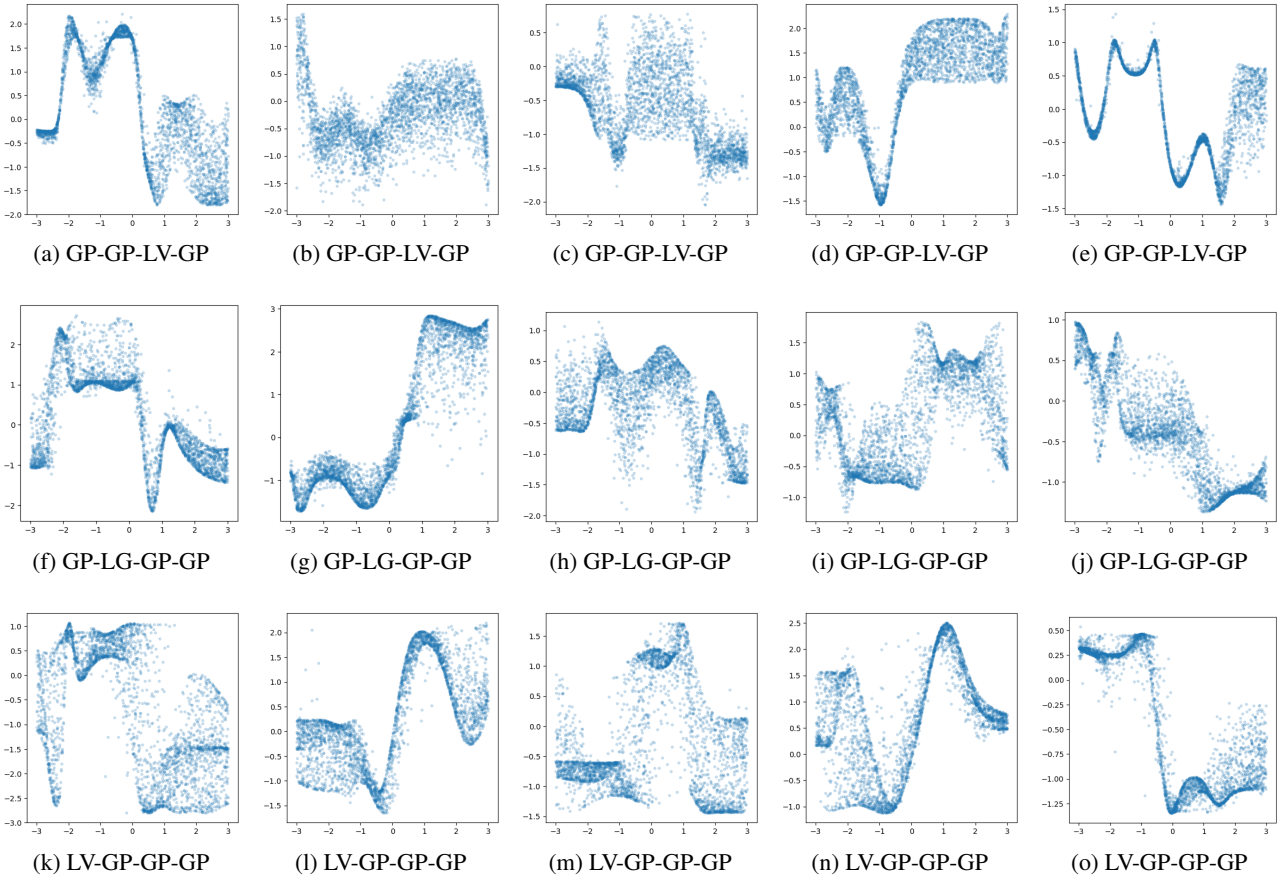
Figure 7: Prior draws from the LV-GP model



(a) GP-LV-GP (b) GP-LV-GP (c) GP-LV-GP (d) GP-LV-GP (e) GP-LV-GP

(f) LV-GP-GP (g) LV-GP-GP (h) LV-GP-GP (i) LV-GP-GP (j) LV-GP-GP

Figure 8: Prior draws from the LV-GP-GP and GP-LV-GP models



(a) GP-GP-LV-GP (b) GP-GP-LV-GP (c) GP-GP-LV-GP (d) GP-GP-LV-GP (e) GP-GP-LV-GP

(f) GP-LG-GP-GP (g) GP-LG-GP-GP (h) GP-LG-GP-GP (i) GP-LG-GP-GP (j) GP-LG-GP-GP

(k) LV-GP-GP-GP (l) LV-GP-GP-GP (m) LV-GP-GP-GP (n) LV-GP-GP-GP (o) LV-GP-GP-GP

Figure 9: Prior draws from LV-GP-GP-GP, GP-LG-GP-GP and GP-GP-LV-GP models

Dataset	N	D	Linear VI	CVAE 50 VI	CVAE 100, 100 VI	GP VI	GP-GP VI	GP-GP-GP VI	GP SGHMC	GP-GP SGHMC	GP-GP-GP SGHMC	LV-GP VI	LV-GP IWVI	LV-GP-GP VI	LV-GP-GP IWVI	LV-GP-GP-GP VI	LV-GP-GP-GP IWVI
challenger	23	4	-1.12 (0.02)	***-0.59 (0.03)***	$-\infty$	-3.22 (0.07)	*-2.04 (0.02)*	*-2.10 (0.03)*	$-\infty$	$-\infty$	*-654 (337)*	*-0.67 (0.01)*	*-0.68 (0.00)*	*-9.99 (3.60)*	*-9.47 (2.19)*	-1.21 (0.01)	*-2.45 (0.04)*
fertility	100	9	**-1.32 (0.01)**	$-\infty$	$-\infty$	-2.51 (0.13)	-1.43 (0.01)	-1.40 (0.01)	-2.19 (0.02)	*-1.67 (0.03)*	*-1.35 (0.02)*	-1.41 (0.01)	-1.40 (0.01)	-1.42 (0.01)	-1.69 (0.01)	**-1.32 (0.00)**	**-1.31 (0.01)**
concreteslump	103	7	0.71 (0.01)	$-\infty$	$-\infty$	1.91 (0.01)	1.55 (0.00)	1.45 (0.00)	0.59 (0.04)	0.10 (0.03)	-0.14 (0.03)	1.93 (0.00)	**1.94 (0.00)**	1.53 (0.00)	1.75 (0.00)	1.43 (0.00)	1.62 (0.00)
autos	159	25	-0.43 (0.00)	$-\infty$	$-\infty$	-0.36 (0.00)	-0.14 (0.01)	-0.14 (0.03)	-10 (1)	-15 (2)	*-23 (4)*	-0.36 (0.01)	-0.35 (0.00)	**-0.06 (0.04)**	-0.34 (0.01)	-0.25 (0.05)	-0.33 (0.02)
servo	167	4	-0.86 (0.00)	*-385 (220)*	$-\infty$	-0.17 (0.00)	-0.19 (0.01)	-0.17 (0.01)	-1.22 (0.21)	-27 (10)	-326 (44)	-0.07 (0.01)	-0.08 (0.01)	-0.19 (0.01)	-0.07 (0.01)	-0.25 (0.01)	**0.03 (0.03)**
breastcancer	194	33	-1.49 (0.01)	$-\infty$	$-\infty$	-1.32 (0.00)	-1.36 (0.00)	-1.34 (0.00)	-10 (1)	*-28 (10)*	*-462 (100)*	**-1.31 (0.00)**	**-1.31 (0.00)**	-1.43 (0.01)	-1.59 (0.10)	-1.38 (0.00)	-1.80 (0.19)
machine	209	7	-0.71 (0.02)	*-114 (78)*	$-\infty$	-0.70 (0.01)	-0.65 (0.02)	-0.61 (0.01)	-2.21 (0.28)	-14 (4)	-74 (13)	-0.75 (0.02)	-0.71 (0.02)	-0.63 (0.01)	**-0.51 (0.01)**	-0.59 (0.01)	**-0.52 (0.03)**
yacht	308	6	-0.14 (0.10)	-7.91 (4.34)	$-\infty$	1.32 (0.02)	1.84 (0.01)	**2.00 (0.06)**	-0.26 (0.79)	-9.70 (1.35)	-16 (2)	1.64 (0.00)	1.65 (0.00)	1.69 (0.09)	1.71 (0.01)	1.77 (0.07)	**2.05 (0.02)**
autompg	392	7	-0.48 (0.00)	*-71 (9)*	$-\infty$	-0.44 (0.01)	-0.57 (0.02)	-0.48 (0.03)	-2.35 (0.22)	-43 (4)	-131 (26)	-0.32 (0.01)	-0.33 (0.01)	-0.38 (0.04)	**-0.24 (0.01)**	-0.65 (0.11)	-0.28 (0.03)
boston	506	13	-0.56 (0.00)	$-\infty$	$-\infty$	0.02 (0.00)	0.07 (0.00)	0.03 (0.01)	0.14 (0.00)	-2.70 (0.70)	-21 (5)	-0.04 (0.00)	-0.07 (0.00)	0.06 (0.00)	0.14 (0.00)	-0.04 (0.01)	**0.17 (0.00)**
forest*	517	12	-1.36 (0.00)	*-47 (15)*	$-\infty$	-1.38 (0.00)	-1.37 (0.00)	-1.37 (0.00)	-1.75 (0.02)	-3.24 (0.48)	-43 (5)	*-0.99 (0.00)*	*-1.00 (0.00)*	***-0.91 (0.00)***	*-1.02 (0.02)*	***-0.92 (0.01)***	***-0.91 (0.01)***
stock	536	11	**-0.16 (0.00)**	$-\infty$	$-\infty$	-0.22 (0.00)	-0.29 (0.00)	-0.26 (0.00)	-0.22 (0.01)	-2.46 (0.41)	-97 (13)	-0.22 (0.00)	-0.22 (0.00)	-0.29 (0.00)	-0.19 (0.00)	-0.23 (0.01)	-0.36 (0.02)
pendulum	630	9	-1.16 (0.00)	$-\infty$	$-\infty$	-0.14 (0.00)	0.22 (0.01)	0.12 (0.08)	-0.11 (0.01)	**0.33 (0.06)**	-5.43 (1.52)	-0.14 (0.00)	-0.14 (0.00)	0.22 (0.01)	**0.33 (0.08)**	**0.25 (0.00)**	0.20 (0.03)
energy	768	8	-0.06 (0.00)	-0.44 (0.31)	$-\infty$	1.71 (0.00)	1.85 (0.00)	2.07 (0.01)	1.75 (0.00)	1.97 (0.01)	1.12 (0.33)	1.86 (0.01)	1.92 (0.01)	2.00 (0.04)	1.96 (0.01)	2.07 (0.01)	**2.28 (0.01)**
concrete	1030	8	-0.97 (0.00)	-26 (10)	$-\infty$	-0.43 (0.00)	-0.45 (0.00)	-0.35 (0.02)	-0.33 (0.01)	-0.88 (0.38)	-1.85 (1.01)	-0.32 (0.00)	-0.32 (0.00)	-0.35 (0.00)	-0.21 (0.00)	-0.22 (0.01)	**-0.12 (0.00)**
solar*	1066	10	-1.54 (0.03)	***0.71 (0.02)***	*0.26 (0.34)*	-1.75 (0.08)	-1.21 (0.02)	-1.20 (0.03)	-1.56 (0.05)	-1.58 (0.08)	-1.65 (0.06)	*0.04 (0.07)*	*0.07 (0.01)*	*0.54 (0.01)*	*0.22 (0.01)*	*0.54 (0.01)*	*0.20 (0.02)*
airfoil	1503	5	-1.11 (0.00)	-0.64 (0.32)	$-\infty$	-0.79 (0.05)	0.08 (0.02)	0.14 (0.03)	-0.90 (0.05)	-0.19 (0.20)	**0.32 (0.09)**	-0.44 (0.03)	-0.36 (0.01)	0.07 (0.00)	0.30 (0.00)	-0.02 (0.03)	**0.34 (0.01)**
winered	1599	11	-1.14 (0.00)	*-9.64 (1.40)*	*-1.86 (0.85)*	-1.09 (0.00)	-1.11 (0.00)	-1.08 (0.00)	-1.11 (0.00)	-1.09 (0.01)	-1.63 (0.09)	-1.07 (0.00)	-1.06 (0.00)	-1.10 (0.00)	***-0.84 (0.01)***	*-1.06 (0.03)*	*-1.31 (0.40)*
gas	2565	128	0.23 (0.00)	-241 (77)	$-\infty$	1.07 (0.00)	**1.60 (0.05)**	**1.69 (0.05)**	0.93 (0.01)	1.20 (0.02)	1.46 (0.05)	**1.69 (0.08)**	1.56 (0.06)	0.70 (0.10)	**1.57 (0.16)**	1.30 (0.14)	**1.61 (0.12)**
skillcraft	3338	19	-0.95 (0.00)	-5.80 (0.29)	$-\infty$	-0.94 (0.00)	-0.94 (0.00)	-0.94 (0.00)	-0.95 (0.00)	-0.99 (0.02)	-1.10 (0.03)	**-0.91 (0.00)**	**-0.91 (0.00)**	-0.92 (0.00)	-0.93 (0.00)	-0.92 (0.00)	-0.94 (0.00)
sml†	4137	26	0.32 (0.00)	-3.79 (0.87)	$-\infty$	1.53 (0.00)	1.72 (0.01)	1.83 (0.01)	1.46 (0.00)	1.71 (0.04)	**1.93 (0.01)**	1.52 (0.00)	1.52 (0.00)	1.79 (0.00)	1.91 (0.00)	**1.92 (0.01)**	**1.97 (0.05)**
winewhite	4898	11	-1.22 (0.00)	-1.37 (0.04)	$-\infty$	-1.14 (0.00)	-1.14 (0.00)	-1.14 (0.00)	-1.13 (0.00)	-1.13 (0.00)	-1.14 (0.01)	-1.13 (0.00)	-1.13 (0.00)	-1.13 (0.00)	**-1.10 (0.00)**	-1.13 (0.00)	**-1.09 (0.00)**
parkinsons	5875	20	-1.30 (0.00)	-1.14 (0.56)	$-\infty$	1.99 (0.00)	2.61 (0.01)	2.75 (0.02)	1.87 (0.02)	2.38 (0.05)	2.41 (0.04)	1.79 (0.02)	1.82 (0.02)	2.36 (0.02)	2.76 (0.02)	2.71 (0.07)	**3.12 (0.05)**
kin8nm†	8192	8	-1.12 (0.00)	-0.01 (0.01)	$-\infty$	-0.29 (0.00)	-0.01 (0.00)	-0.00 (0.00)	-0.30 (0.00)	-0.03 (0.01)	-0.00 (0.00)	-0.29 (0.00)	-0.29 (0.00)	-0.02 (0.00)	-0.00 (0.00)	-0.00 (0.00)	**0.03 (0.00)**
pumadyn32nm	8192	32	-1.44 (0.00)	-2.10 (0.09)	$-\infty$	0.08 (0.00)	0.11 (0.01)	0.11 (0.00)	**0.11 (0.00)**	**0.11 (0.00)**	0.10 (0.00)	-1.44 (0.00)	-1.44 (0.00)	0.10 (0.01)	*-0.62 (0.25)*	0.11 (0.00)	0.11 (0.00)
power*	9568	4	-0.43 (0.01)	-0.44 (0.08)	-2.14 (0.06)	-0.65 (0.04)	-0.75 (0.03)	-0.80 (0.02)	-0.58 (0.03)	-1.41 (0.00)	-1.41 (0.00)	-0.39 (0.04)	-0.23 (0.02)	-0.36 (0.04)	*-0.28 (0.05)*	-0.25 (0.06)	***-0.11 (0.06)***
naval	11934	14	-0.63 (0.00)	3.07 (0.08)	3.39 (0.12)	**4.52 (0.02)**	4.43 (0.03)	4.35 (0.03)	3.17 (0.03)	-1.41 (0.00)	-1.41 (0.00)	*4.19 (0.01)*	*4.24 (0.01)*	4.36 (0.01)	**4.52 (0.02)**	4.27 (0.02)	4.41 (0.02)
pol‡	15000	26	-1.10 (0.00)	-0.05 (0.08)	-411 (19)	0.48 (0.00)	1.51 (0.01)	1.45 (0.01)	0.46 (0.00)	1.32 (0.03)	1.31 (0.07)	*0.37 (0.08)*	*-0.50 (0.00)*	*2.34 (0.02)*	***2.63 (0.01)***	1.47 (0.01)	**2.72 (0.10)**
elevators	16599	18	-0.74 (0.00)	-0.28 (0.00)	-33 (1)	-0.44 (0.00)	-0.41 (0.01)	-0.40 (0.00)	-0.46 (0.00)	-0.40 (0.01)	-0.39 (0.01)	-0.37 (0.00)	-0.36 (0.00)	-0.29 (0.00)	-0.27 (0.00)	-0.28 (0.00)	**-0.27 (0.00)**
bike‡	17379	17	-0.76 (0.00)	*3.90 (0.01)*	**4.47 (0.02)**	0.82 (0.01)	3.49 (0.01)	3.68 (0.01)	1.06 (0.01)	1.21 (0.95)	2.77 (0.10)	*2.48 (0.01)*	*2.66 (0.01)*	3.48 (0.00)	3.75 (0.01)	3.72 (0.01)	3.95 (0.01)
kin40k†	40000	8	-1.43 (0.00)	0.65 (0.01)	**1.24 (0.13)**	0.02 (0.00)	0.84 (0.00)	1.17 (0.00)	-0.02 (0.00)	-1.43 (0.00)	-1.43 (0.00)	0.04 (0.00)	0.05 (0.00)	0.87 (0.01)	0.93 (0.00)	1.15 (0.00)	**1.27 (0.00)**
protein	45730	9	-1.25 (0.00)	*-0.78 (0.02)*	*-0.57 (0.00)*	-1.06 (0.00)	-0.98 (0.00)	-0.95 (0.00)	-1.08 (0.00)	-1.10 (0.07)	-0.95 (0.00)	*-0.85 (0.00)*	*-0.80 (0.00)*	*-0.70 (0.00)*	*-0.61 (0.00)*	*-0.68 (0.00)*	***-0.57 (0.00)***
tamielectric	45781	3	-1.43 (0.00)	*-1.31 (0.00)*	***-1.31 (0.00)***	-1.43 (0.00)	-1.43 (0.00)	-1.43 (0.00)	-1.43 (0.00)	-1.43 (0.00)	-1.43 (0.00)	*-1.31 (0.00)*	*-1.31 (0.00)*	*-1.31 (0.00)*	*-1.31 (0.00)*	*-1.31 (0.00)*	*-1.31 (0.00)*
keggdirected‡	48827	20	-0.34 (0.15)	*0.65 (0.32)*	*1.59 (0.14)*	0.16 (0.01)	0.21 (0.01)	0.26 (0.02)	0.26 (0.02)	0.45 (0.02)	0.53 (0.04)	*1.24 (0.06)*	*2.03 (0.01)*	*1.84 (0.05)*	*2.23 (0.01)*	*1.92 (0.02)*	***2.26 (0.01)***
slice†	53500	385	-0.47 (0.00)	-27 (1)	-358 (15)	0.86 (0.00)	1.80 (0.00)	1.86 (0.00)	1.02 (0.00)	1.84 (0.02)	1.93 (0.01)	0.85 (0.00)	0.90 (0.00)	1.80 (0.00)	1.88 (0.00)	1.86 (0.00)	**2.02 (0.00)**
keggundirected*	63608	27	0.28 (0.00)	*-344 (39)*	$-\infty$	0.06 (0.01)	0.06 (0.01)	0.07 (0.01)	-0.03 (0.02)	-0.14 (0.23)	0.10 (0.01)	*-75 (4)*	*-4.21 (0.33)*	*-64 (6)*	*2.37 (0.29)*	*-116 (17)*	***2.98 (0.21)***
3droad	434874	3	-1.41 (0.00)	*-0.85 (0.00)*	*-0.53 (0.01)*	-0.79 (0.00)	-0.61 (0.01)	-0.58 (0.01)	-1.15 (0.06)	-1.42 (0.00)	-1.42 (0.00)	*-0.71 (0.00)*	*-0.65 (0.00)*	*-0.69 (0.00)*	*-0.55 (0.01)*	*-0.61 (0.00)*	***-0.48 (0.00)***
song	515345	90	-1.28 (0.00)	-1.11 (0.00)	-1.09 (0.00)	-1.19 (0.00)	-1.18 (0.00)	-1.18 (0.00)	-1.19 (0.00)	-1.17 (0.00)	-1.17 (0.00)	-1.15 (0.00)	-1.14 (0.00)	-1.13 (0.00)	*-1.10 (0.00)*	-1.12 (0.00)	***-1.07 (0.00)***
buzz	583250	77	-1.56 (0.04)	0.01 (0.00)	**0.07 (0.00)**	-0.24 (0.00)	-0.15 (0.00)	-0.15 (0.00)	-0.92 (0.02)	-0.48 (0.01)	-0.43 (0.03)	-0.22 (0.01)	-0.23 (0.01)	-0.09 (0.01)	-0.04 (0.03)	-0.09 (0.01)	*0.04 (0.00)*
nytaxi	1420068	8	-1.53 (0.01)	*-1.64 (0.04)*	*-1.90 (0.04)*	-1.42 (0.01)	-1.44 (0.01)	-1.42 (0.01)	-1.46 (0.01)	-1.78 (0.00)	-1.77 (0.00)	*-1.09 (0.03)*	***-0.90 (0.04)***	*-1.57 (0.04)*	*-1.03 (0.04)*	*-1.74 (0.05)*	***-0.95 (0.07)***
houseelectric‡	2049280	11	-0.52 (0.00)	1.41 (0.00)	1.48 (0.01)	1.37 (0.00)	1.41 (0.00)	1.47 (0.01)	0.01 (0.04)	-1.43 (0.00)	-1.43 (0.00)	*1.65 (0.03)*	*1.82 (0.01)*	1.66 (0.05)	*2.01 (0.00)*	1.77 (0.06)	***2.03 (0.00)***
Median difference from GP baseline			-0.38	-2.06	-6016.67	0	0.06	0.09	-0.02	-0.02	-0.28	0.04	0.05	0.12	0.26	0.18	0.32
Average ranks			4.99 (0.25)	5.32 (0.31)	4.35 (0.38)	6.80 (0.21)	8.65 (0.20)	9.63 (0.20)	5.57 (0.19)	5.57 (0.24)	5.21 (0.28)	8.94 (0.22)	9.36 (0.22)	9.82 (0.17)	12.04 (0.18)	10.71 (0.19)	13.03 (0.20)

Table 2: Test log likelihood values over 5 splits (standard errors), including the CVAE models and models with Stochastic Gradient HMC Havasi et al. (2018). Results less than -1000 are reported as $-\infty$. The numbers after the CVAE models are the number of hidden units. The SGHMC results were run with identical models as the corresponding VI methods, Hyperparameter optimization was performed using a random sample from the last 100 iterations, as described in (Havasi et al., 2018). Posterior sample were take with 2000 samples, with a thinning factor of 5 (i.e. a chain of length 10000). We included also a single layer GP using the SGHMC approach for comparison. The discrepancy between the variational approach and the SGHMC is attributable to several factors (note that the variational approach is optimal in this case): different batch sizes (512 in this work compared to 10000 in Havasi et al. (2018)); different hyperparameter initializations (for example, a much longer initial lengthscales is used in Havasi et al. (2018)); different learning rate schedules (decaying learning rate in this work compared to fixed in Havasi et al. (2018)); the use of natural gradients in this work for the final layer, as opposed to optimizing with Adam optimizer, as done in Havasi et al. (2018); Salimbeni and Deisenroth (2017); insufficient mixing in the posterior chain.

Dataset	N	D	Linear VI	CVAE 50 VI	CVAE 100, 100 VI	GP VI	GP-GP VI	GP-GP-GP VI	GP SGHMC	GP-GP SGHMC	GP-GP-GP SGHMC	LV-GP VI	LV-GP IWVI	LV-GP-GP VI	LV-GP-GP IWVI	LV-GP-GP-GP VI	LV-GP-GP-GP IWVI
challenger	23	4		0.5191		0.9990	0.8870	0.7739		0.9823	0.9099	0.7001	0.7115	0.8466	0.8484		0.7187
fertility	100	9		0.9408	0.8353					0.9598	0.9341				0.9928		0.9969
concreteslump	103	7								0.9970	0.9922						
autos	159	25		0.9823	0.8798					0.9903	0.9874				0.9990		0.9990
servo	167	4		0.8358						0.9989	0.9975	0.9990	0.9990				0.9988
breastcancer	194	33								0.9787	0.9814				0.9911		0.9943
machine	209	7		0.9181	0.8967					0.9974	0.9989						
yacht	308	6		0.9987											0.9989		0.9990
autompg	392	7		0.9097	0.9948					0.9989	0.9987	0.9990	0.9988	0.9987	0.9985		0.9988
boston	506	13		0.9916							0.9989	0.9989	0.9979	0.9989	0.9980		0.9981
forest*	517	12		0.8453	0.7145					0.9981	0.9964	0.7034	0.7074	0.7470	0.7398	0.7608	0.7642
stock	536	11															
pendulum	630	9													0.9989		0.9988
energy	768	8											0.9987				0.9990
concrete	1030	8		0.9986	0.9782										0.9990		0.9989
solar*	1066	10		0.3028	0.3228							0.3380	0.3509	0.2728	0.2945	0.2557	0.2883
airfoil	1503	5		0.9989	0.9986										0.9988		0.9988
winered	1599	11		0.9790	0.6980										0.9234	0.9751	0.8473
gas	2565	128		0.9984								0.9983	0.9935	0.9989	0.9983	0.9990	0.9986
skillcraft	3338	19										0.9989	0.9983	0.9989	0.9988	0.9990	0.9989
sml†	4137	26															
winewhite	4898	11													0.9990		0.9990
parkinsons	5875	20		0.9989													
kin8nm†	8192	8															
pumadyn32nm	8192	32													0.9770		
power*	9568	4		0.9964								0.9963	0.9926	0.9964	0.9852	0.9963	0.9855
naval	11934	14										0.9649	0.9332				
pol‡	15000	26		0.9970								0.8482	0.7214	0.9839	0.9508		0.9980
elevators	16599	18		0.9979								0.9986	0.9977	0.9987	0.9976	0.9988	0.9979
bike‡	17379	17		0.9815								0.9657	0.9337	0.9990	0.9990		
kin40k†	40000	8		0.9990													
protein	45730	9		0.8839	0.8800							0.9117	0.9041	0.8904	0.8761	0.8840	0.8691
tamielectric	45781	3		0.9551	0.9502							0.9558	0.9562	0.9550	0.9549	0.9551	0.9549
keggdirected‡	48827	20		0.8279	0.5350							0.7498	0.5445	0.6839	0.5187	0.7039	0.5001
slice†	53500	385										0.9990	0.9983		0.9988		0.9988
keggundirected*	63608	27		0.4804	0.9882							0.5287	0.3281	0.5511	0.2049	0.9295	0.2322
3droad	434874	3		0.9421	0.9479							0.9432	0.9271	0.9650	0.9467	0.9744	0.9464
song	515345	90		0.9936	0.9957							0.9983	0.9932	0.9969	0.9893	0.9963	0.9835
buzz	583250	77		0.9952	0.9909									0.9988	0.9938	0.9984	0.9838
nytaxi	1420068	8		0.9608	0.9422							0.9698	0.9274	0.9515	0.9019	0.9467	0.8981
houseelectric‡	2049280	11										0.9830	0.9676	0.9962	0.9671	0.9933	0.9743

Table 3: Median Shapiro–Wilk test statistic for the test points. Blanks have values of one (perfectly Gaussian). Smaller values indicate more evidence for non-Gaussian marginal distributions.

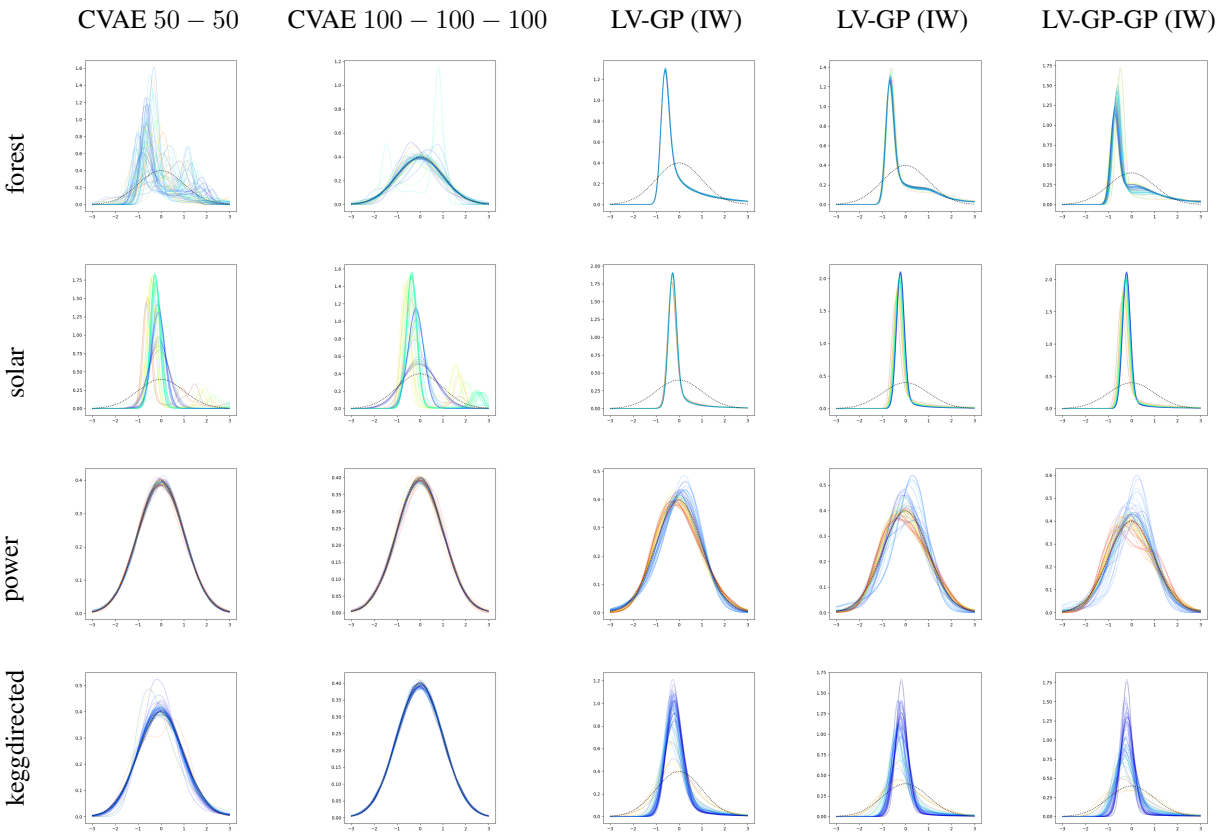


Figure 10: Posterior marginals, re-scaled to zero mean and unit standard deviation, with the Gaussian density marked as a dotted curve. Colors are according to the principal component of the inputs. These are the four datasets highlighted in the main text for being prominent examples benefiting from latent variables. In these cases the additional depth of model leads to more non-Gaussian marginals.

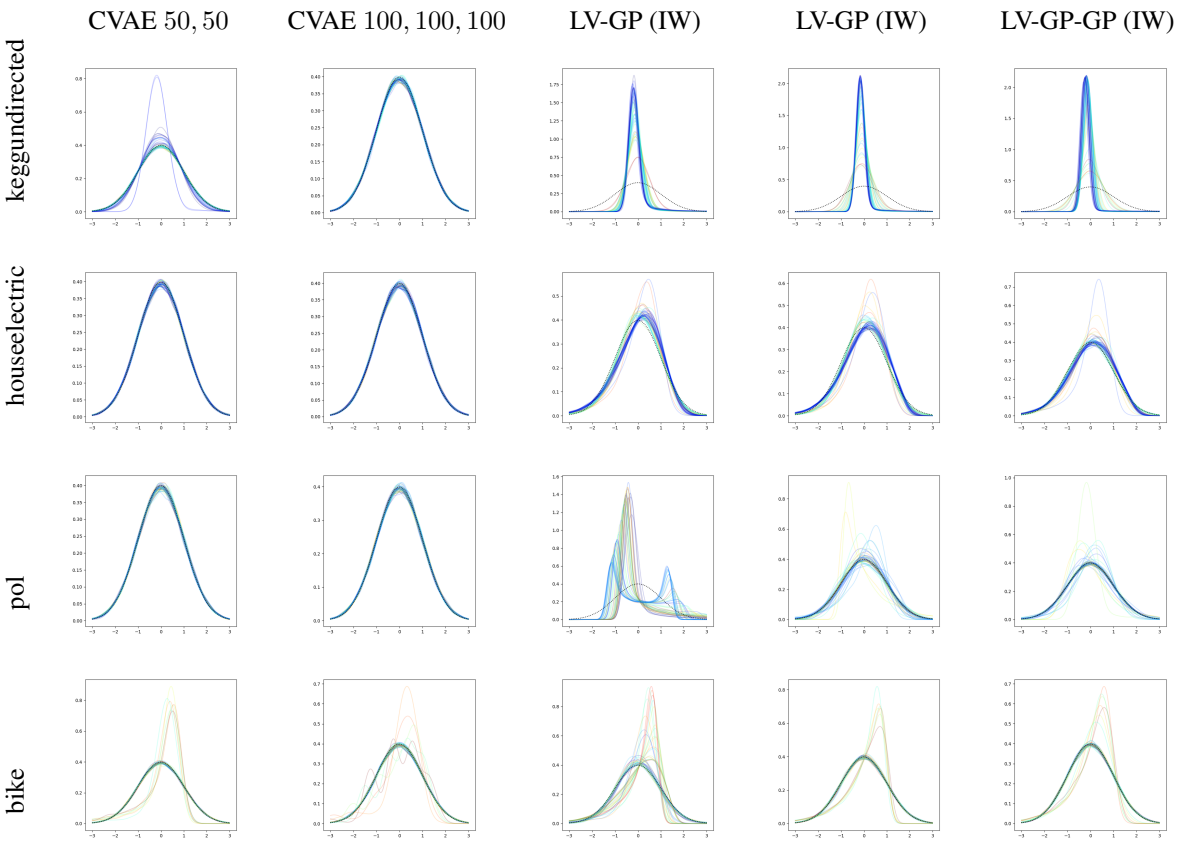


Figure 11: As Figure 10, but for the four datasets highlighted in the main text for benefiting from both latent variables and depth. Note that, unlike in Figure 10, for the 'pol' and 'bike' data the deeper models actually have *simpler* marginals. This is because the deep model has more complexity in the mapping, so can more closely fit the data, whereas the simple model must explain the data with a complex noise distribution.