Supplementary material for deep learning with differential Gaussian process flows

A. Derivation of the stochastic variational inference

The differential Gaussian process is a combination of a conventional prediction GP $g(\cdot)$ with an SDE flow GP $\mathbf{f}(\cdot)$ fully parameterised by $\mathbf{Z}, \mathbf{U}$ as well as kernel parameters θ. We turn to variational inference to estimate posterior approximations $q(\mathbf{U_f})$ and $q(\mathbf{u}_g)$ for both models.

Exact inference of Gaussian processes has a limiting complexity of $\mathcal{O}(N^3)$. Instead, we apply stochastic variational inference (SVI) (Hensman et al., 2015), which has been demonstrated to scale GP's up to a billion data points (Salimbeni and Deisenroth, 2017). We here summarise the SVI procedure following Hensman et al. (2015).

We start with the joint density of the augmented system

$$p(\mathbf{y}, \mathbf{g}, \mathbf{u}_g, \mathbf{X}_T, \mathbf{f}, \mathbf{U_f}) = \underbrace{p(\mathbf{y}|\mathbf{g})}_{\text{likelihood}} \underbrace{p(\mathbf{g}|\mathbf{u}_g, \mathbf{X}_T)p(\mathbf{u}_g)}_{g(\mathbf{x})\text{ GP prior}} \underbrace{p(\mathbf{X}_T|\mathbf{f})}_{\text{SDE}} \underbrace{p(\mathbf{f}|\mathbf{U_f})p(\mathbf{U_f})}_{\mathbf{f}(\mathbf{x})\text{ GP prior}},$$

where we have augmented the predictor function g with M inducing locations $\mathbf{Z}_g = (\mathbf{z}_{g1}, \ldots, \mathbf{z}_{gM})$ with associated inducing function values $g(\mathbf{z}) = u$ in a vector $\mathbf{u}_g = (u_{g1}, \ldots, u_{gM})^T \in \mathbb{R}^M$ with a GP prior. The conditional distribution is (Titsias, 2009)

$$p(\mathbf{g}|\mathbf{u}_g, \mathbf{X}_T) = \mathcal{N}(\mathbf{g}|\mathbf{Q}_T\mathbf{u}_g, \mathbf{K}_{\mathbf{X}_T\mathbf{X}_T} - \mathbf{Q}_T\mathbf{K}_{\mathbf{Z}_g\mathbf{Z}_g}\mathbf{Q}_T^T) \tag{1}$$

$$p(\mathbf{u}_g) = \mathcal{N}(\mathbf{u}_g|\mathbf{0}, \mathbf{K}_{\mathbf{Z}_g\mathbf{Z}_g}), \tag{2}$$

where we denote $\mathbf{Q}_T = \mathbf{K}_{\mathbf{X}_T\mathbf{Z}_g}\mathbf{K}^{-1}_{\mathbf{Z}_g\mathbf{Z}_g}$.

Similarly, the warping function $\mathbf{f}$ is augmented with inducing variables $\mathbf{U_f} = (\mathbf{u_f}_1, \ldots, \mathbf{u_f}_D)$ and inducing locations $\mathbf{Z_f}$. In addition, the inducing variables $\mathbf{U_f}$ are also given a GP prior.

$$p(\mathbf{f}|\mathbf{U_f}) = \mathcal{N}(\mathbf{f}|\mathbf{R}\mathbf{U_f}, \mathbf{K_{XX}} - \mathbf{R}\mathbf{K}_{\mathbf{Z_f}\mathbf{Z_f}}\mathbf{R}^T) \tag{3}$$

$$p(\mathbf{U_f}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{u_f}_d|\mathbf{0}, \mathbf{K}_{\mathbf{Z_f}\mathbf{Z_f}}). \tag{4}$$

where we denote $\mathbf{R} = \mathbf{K}_{\mathbf{X}\mathbf{Z_f}}\mathbf{K}^{-1}_{\mathbf{Z_f}\mathbf{Z_f}}$.

The joint distribution contains the likelihood term, the two GP priors, and the the key component of the state distribution $p(\mathbf{X}_T|\mathbf{f}) := p(\mathbf{X}_T; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{X}_0)$, which follows the intractable Fokker-Planck equation

$$\frac{\partial p_t(\mathbf{x})}{\partial t} = -\nabla^T_{\mathbf{x}}(\boldsymbol{\mu}(\mathbf{x})p_t(\mathbf{x})) + \frac{1}{2}\nabla^T_{\mathbf{x}}(p_t(\mathbf{x})\Sigma(\mathbf{x}))\nabla_{\mathbf{x}}, \tag{5}$$

and *all* solutions to the SDE of equation

$$d\mathbf{x}_t = \boldsymbol{\mu}(\mathbf{x}_t)dt + \sqrt{\Sigma(\mathbf{x}_t)}dW_t \tag{6}$$

$$\boldsymbol{\mu}(\mathbf{x}_t) = \mathbf{R}\mathbf{U_f} \tag{7}$$

$$\boldsymbol{\Sigma}(\mathbf{x}_t) = \mathbf{K_{xx}} - \mathbf{R}\mathbf{K}_{\mathbf{Z_f}\mathbf{Z_f}}\mathbf{R} \tag{8}$$

We consider optimizing the marginal likelihood

$$\log p(\mathbf{y}) = \log \mathbb{E}_{p(\mathbf{g}|\mathbf{X}_T)p(\mathbf{X}_T)} p(\mathbf{y}|\mathbf{g}), \tag{9}$$

$$p(\mathbf{g}|\mathbf{X}_T) = \int p(\mathbf{g}|\mathbf{u}_g, \mathbf{X}_T) p(\mathbf{u}_g) d\mathbf{u}_g \tag{10}$$

$$p(\mathbf{X}_T) = \iint p(\mathbf{X}_T|\mathbf{f}) p(\mathbf{f}|\mathbf{U}_\mathbf{f}) p(\mathbf{U}_\mathbf{f}) d\mathbf{f} d\mathbf{U}_\mathbf{f}, \tag{11}$$

with no tractable solution due to the FPK state distribution $p(\mathbf{X}_T)$.

A variational lower bound for the evidence (9) without the state distributions has already been considered by Hensman et al. (2015). We propose to include the state distributions by simulating Monte Carlo state trajectories.

We propose a complete variational posterior approximation over both $\mathbf{f}$ and g,

$$q(\mathbf{g}, \mathbf{u}_g, \mathbf{X}_T, \mathbf{f}, \mathbf{U}_\mathbf{f}) = p(\mathbf{g}|\mathbf{u}_g, \mathbf{X}_T) q(\mathbf{u}_g) p(\mathbf{X}_T|\mathbf{f}) p(\mathbf{f}|\mathbf{U}_\mathbf{f}) q(\mathbf{U}_\mathbf{f}) \tag{12}$$

$$q(\mathbf{u}_g) = \mathcal{N}(\mathbf{u}_g|\mathbf{m}_g, \mathbf{S}_g) \tag{13}$$

$$q(\mathbf{U}_\mathbf{f}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{u}_{\mathbf{f}d}|\mathbf{m}_{\mathbf{f}d}, \mathbf{S}_{\mathbf{f}d}), \tag{14}$$

where $\mathbf{M}_\mathbf{f} = (\mathbf{m}_{\mathbf{f}1}, \ldots, \mathbf{m}_{\mathbf{f}D})$ and $\mathbf{S}_\mathbf{f} = (\mathbf{S}_{\mathbf{f}1}, \ldots, \mathbf{S}_{\mathbf{f}D})$ collect the dimension-wise inducing parameters. We continue by marginalizing out inducing variables $\mathbf{u}_g$ and $\mathbf{U}_\mathbf{f}$ from the above joint distribution arriving at the joint variational posterior

$$q(\mathbf{g}, \mathbf{X}_T, \mathbf{f}) = q(\mathbf{g}|\mathbf{X}_T) p(\mathbf{X}_T|\mathbf{f}) q(\mathbf{f}), \tag{15}$$

where

$$q(\mathbf{g}|\mathbf{X}_T) = \int p(\mathbf{g}|\mathbf{u}_g, \mathbf{X}_T) q(\mathbf{u}_g) d\mathbf{u}_g \tag{16}$$

$$= \mathcal{N}(\mathbf{g}|\mathbf{Q}_T \mathbf{m}_g, \mathbf{K}_{\mathbf{X}_T\mathbf{X}_T} + \mathbf{Q}_T(\mathbf{S}_g - \mathbf{K}_{\mathbf{Z}_g\mathbf{Z}_g})\mathbf{Q}_T^T) \tag{17}$$

$$q(\mathbf{f}) = \int p(\mathbf{f}|\mathbf{U}_\mathbf{f}) q(\mathbf{U}_\mathbf{f}) d\mathbf{U}_\mathbf{f} \tag{18}$$

$$= \mathcal{N}(\mathbf{f}|\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$$

$$\boldsymbol{\mu}_q = \mathbf{Q}_\mathbf{f} \mathbf{M}_\mathbf{f} \tag{19}$$

$$\boldsymbol{\Sigma}_q = \mathbf{K}_{\mathbf{X}\mathbf{X}} + \mathbf{Q}_\mathbf{f}(\mathbf{S}_\mathbf{f} - \mathbf{K}_{\mathbf{Z}_\mathbf{f}\mathbf{Z}_\mathbf{f}})\mathbf{Q}_\mathbf{f}^T \tag{20}$$

where $\mathbf{Q}_\mathbf{f} = \mathbf{K}_{\mathbf{X}\mathbf{Z}_\mathbf{f}} \mathbf{K}_{\mathbf{Z}_\mathbf{f}\mathbf{Z}_\mathbf{f}}^{-1}$. We plug the derived variational posterior drift $\boldsymbol{\mu}_q$ and diffusion $\boldsymbol{\Sigma}_q$ estimates to the SDE to arrive at the final *variational SDE flow*

$$d\mathbf{x}_t = \boldsymbol{\mu}_q(\mathbf{x}_t) dt + \sqrt{\boldsymbol{\Sigma}_q(\mathbf{x}_t)} dW_t, \tag{21}$$

which conveniently encodes the variational approximation of $\mathbf{f}$.

Now the lower bound for our differential deep GP model can be written as

$$\log p(\mathbf{y}) \geq \int q(\mathbf{g}, \mathbf{u}_g, \mathbf{X}_T, \mathbf{f}, \mathbf{U}_\mathbf{f}) \log \frac{p(\mathbf{y}, \mathbf{g}, \mathbf{u}_g, \mathbf{X}_T, \mathbf{f}, \mathbf{U}_\mathbf{f})}{q(\mathbf{g}, \mathbf{u}_g, \mathbf{X}_T, \mathbf{f}, \mathbf{U}_\mathbf{f})} d\mathbf{g} d\mathbf{u}_g d\mathbf{X}_T d\mathbf{f} d\mathbf{U}_\mathbf{f} \tag{22}$$

$$\geq \int p(\mathbf{g}|\mathbf{u}_g, \mathbf{X}_T) q(\mathbf{u}_g) p(\mathbf{X}_T|\mathbf{f}) p(\mathbf{f}|\mathbf{U}_\mathbf{f}) q(\mathbf{U}_\mathbf{f}) \log \frac{p(\mathbf{y}|\mathbf{g}) p(\mathbf{u}_g) p(\mathbf{U}_\mathbf{f})}{q(\mathbf{u}_g) q(\mathbf{U}_\mathbf{f})} d\mathbf{g} d\mathbf{u}_g d\mathbf{X}_T d\mathbf{f} d\mathbf{U}_\mathbf{f} \tag{23}$$

$$\geq \int q(\mathbf{g}|\mathbf{X}_T) q(\mathbf{X}_T) \log p(\mathbf{y}|\mathbf{g}) d\mathbf{g} d\mathbf{X}_T - \mathrm{KL}[q(\mathbf{u}_g)||p(\mathbf{u}_g)] - \mathrm{KL}[q(\mathbf{U}_\mathbf{f})||p(\mathbf{U}_\mathbf{f})] \tag{24}$$

$$\gtrapprox \sum_{i=1}^{N} \left\{ \frac{1}{S} \sum_{s=1}^{S} \underbrace{\mathbb{E}_{q(\mathbf{g}|\mathbf{x}_{i,T}^{(s)})} \log p(y_i|g_i)}_{\text{variational expected likelihood}} - \underbrace{\mathrm{KL}[q(\mathbf{u}_g)||p(\mathbf{u}_g)]}_{g(x) \text{ prior divergence}} - \underbrace{\mathrm{KL}[q(\mathbf{U}_\mathbf{f})||p(\mathbf{U}_\mathbf{f})]}_{\mathbf{f}(x) \text{ prior divergence}} \right\}. \tag{25}$$

References

J. Hensman, A. Matthews, and Z. Ghahramani. Scalable variational Gaussian process classification. In *Artificial Intelligence and Statistics*, pages 351–360, 2015.

Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep gaussian processes. In *Advances in Neural Information Processing Systems*, pages 4591–4602, 2017.

M. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009.

B. Regression and classification benchmarks

		boston	energy	concrete	wine_red	kin8mn	power	naval	protein
	N	506	768	1,030	1,599	8,192	9,568	11,934	45,730
	D	13	8	8	22	8	4	26	9
Linear		4.24(0.16)	2.88(0.05)	10.54(0.13)	0.65(0.01)	0.20(0.00)	4.51(0.03)	0.01(0.00)	5.21(0.02)
BNN	$L = 2$	3.01(0.18)	1.80(0.05)	5.67(0.09)	0.64(0.01)	0.10(0.00)	4.12(0.03)	0.01(0.00)	4.73(0.01)
Sparse GP	$M = 100$	2.87(0.15)	0.78(0.02)	5.97(0.11)	0.63(0.01)	0.09(0.00)	3.91(0.03)	**0.00**(0.00)	4.43(0.03)
	$M = 500$	2.73(0.12)	**0.47**(0.02)	5.53(0.12)	**0.62**(0.01)	0.08(0.00)	3.79(0.03)	**0.00**(0.00)	4.10(0.03)
Deep GP $M = 100$	$L = 2$	2.90(0.17)	**0.47**(0.01)	5.61(0.10)	0.63(0.01)	**0.06**(0.00)	3.79(0.03)	**0.00**(0.00)	4.00(0.03)
	$L = 3$	2.93(0.16)	0.48(0.01)	5.64(0.10)	0.63(0.01)	**0.06**(0.00)	3.73(0.04)	**0.00**(0.00)	**3.81**(0.04)
	$L = 4$	2.90(0.15)	0.48(0.01)	5.68(0.10)	0.63(0.01)	**0.06**(0.00)	3.71(0.04)	**0.00**(0.00)	**3.74**(0.04)
	$L = 5$	2.92(0.17)	**0.47**(0.01)	5.65(0.10)	0.63(0.01)	**0.06**(0.00)	3.68(0.03)	**0.00**(0.00)	**3.72**(0.04)
DiffGP $M = 100$	$T = 1.0$	2.80(0.13)	0.49(0.02)	**5.32**(0.10)	0.63(0.01)	**0.06**(0.00)	3.76(0.03)	**0.00**(0.00)	4.04(0.04)
	$T = 2.0$	**2.68**(0.10)	0.48(0.02)	**4.96**(0.09)	0.63(0.01)	**0.06**(0.00)	3.72(0.03)	**0.00**(0.00)	4.00(0.04)
	$T = 3.0$	**2.69**(0.14)	**0.47**(0.02)	**4.76**(0.12)	0.63(0.01)	**0.06**(0.00)	3.68(0.03)	**0.00**(0.00)	3.92(0.04)
	$T = 4.0$	**2.67**(0.13)	0.49(0.02)	**4.65**(0.12)	0.63(0.01)	**0.06**(0.00)	**3.66**(0.03)	**0.00**(0.00)	3.89(0.04)
	$T = 5.0$	**2.58**(0.12)	0.50(0.02)	**4.56**(0.12)	0.63(0.01)	**0.06**(0.00)	**3.65**(0.03)	**0.00**(0.00)	3.87(0.04)

Table 1: Test RMSE values of 8 benchmark datasets (reproduced from from Salimbeni & Deisenroth 2017). Uses random 90% / 10% training and test splits, repeated 20 times.

		boston	energy	concrete	wine_red	kin8mn	power	naval	protein
	N	506	768	1,030	1,599	8,192	9,568	11,934	45,730
	D	13	8	8	22	8	4	26	9
Linear		-2.89(0.03)	-2.48(0.02)	-3.78(0.01)	-0.99(0.01)	0.18(0.01)	-2.93(0.01)	3.73(0.00)	-3.07(0.00)
BNN	$L = 2$	-2.57(0.09)	-2.04(0.02)	-3.16(0.02)	-0.97(0.01)	0.90(0.01)	-2.84(0.01)	3.73(0.01)	-2.97(0.00)
Sparse GP	$M = 100$	-2.47(0.05)	-1.29(0.02)	-3.18(0.02)	-0.95(0.01)	0.63(0.01)	-2.75(0.01)	6.57(0.15)	-2.91(0.00)
	$M = 500$	-2.40(0.07)	**-0.63**(0.03)	-3.09(0.02)	**-0.93**(0.01)	1.15(0.00)	-2.75(0.01)	**7.01**(0.05)	-2.83(0.00)
Deep GP $M = 100$	$L = 2$	-2.47(0.05)	-0.73(0.02)	-3.12(0.01)	-0.95(0.01)	1.34(0.01)	-2.75(0.01)	6.76(0.19)	-2.81(0.00)
	$L = 3$	-2.49(0.05)	-0.75(0.02)	-3.13(0.01)	-0.95(0.01)	1.37(0.01)	-2.74(0.01)	6.62(0.18)	**-2.75**(0.00)
	$L = 4$	-2.48(0.05)	-0.76(0.02)	-3.14(9 .01)	-0.95(0.01)	**1.38**(0.01)	-2.74(0.01)	6.61(0.17)	**-2.73**(0.00)
	$L = 5$	-2.49(0.05)	-0.74(0.02)	-3.13(0.01)	-0.95(0.01)	**1.38**(0.01)	-2.73(0.01)	6.41(0.28)	**-2.71**(0.00)
DiffGP $M = 100$	$T = 1.0$	**-2.36**(0.05)	-0.65(0.03)	**-3.05**(0.02)	-0.96(0.01)	1.36(0.01)	-2.75(0.01)	6.58(0.02)	-2.79(0.04)
	$T = 2.0$	**-2.32**(0.04)	**-0.63**(0.03)	**-2.96**(0.02)	-0.97(0.02)	1.37(0.00)	-2.74(0.01)	6.26(0.03)	-2.78(0.04)
	$T = 3.0$	**-2.31**(0.05)	**-0.63**(0.02)	**-2.93**(0.04)	-0.97(0.02)	1.37(0.01)	**-2.72**(0.01)	6.00(0.03)	-2.79(0.00)
	$T = 4.0$	**-2.33**(0.06)	-0.65(0.02)	**-2.91**(0.04)	-0.98(0.02)	1.37(0.01)	**-2.72**(0.01)	5.86(0.02)	-2.78(0.00)
	$T = 5.0$	**-2.30**(0.05)	-0.66(0.02)	**-2.90**(0.05)	-0.98(0.02)	1.36(0.01)	**-2.72**(0.01)	5.78(0.02)	-2.77(0.00)

Table 2: Test log likelihood values of 8 benchmark datasets (reproduced from from Salimbeni & Deisenroth 2017)

		SUSY	HIGGS
	N	5,500,000	11,000,000
	D	18	28
DNN		0.876	**0.885**
Sparse GP	$M = 100$	0.875	0.785
	$M = 500$	0.876	0.794
Deep GP $M = 100$	$L = 2$	0.877	0.830
	$L = 3$	0.877	0.837
	$L = 4$	0.877	0.841
	$L = 5$	0.877	**0.846**
DiffGP $M = 100$	$t = 1.0$	**0.878**	0.840
	$t = 3.0$	**0.878**	0.841
	$t = 5.0$	**0.878**	0.842
DiffGP Temporal $M_s = 100$ $M_t = 3$	$t = 5.0$	**0.878**	**0.846**

Table 3: Test AUC values for large-scale classification datasets. Uses random 90% / 10% training and test splits.