Deep learning with differential Gaussian process flows

Pashupati Hegde **Markus Heinonen** **Harri Lähdesmäki** **Samuel Kaski**

Helsinki Institute for Information Technology HIIT
Department of Computer Science, Aalto University

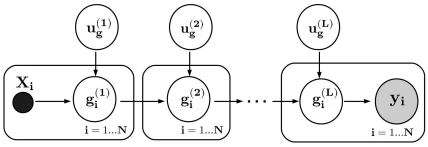
Abstract

We propose a novel deep learning paradigm of differential flows that learn stochastic differential equation transformations of inputs prior to a standard classification or regression task. The key property of differential Gaussian processes is the warping of inputs through infinitely deep, but infinitesimal, differential fields, that generalize discrete layers into a stochastic dynamical system. We demonstrate promising results on various regression benchmark datasets.

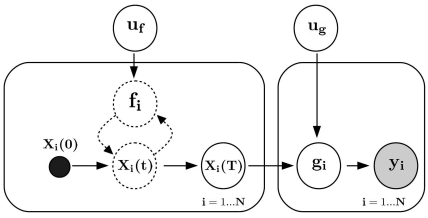
1 Introduction

Gaussian processes are a family of flexible kernel function distributions [Rasmussen and Williams, 2006]. The capacity of kernel models is inherently determined by the function space induced by the choice of the kernel, where standard stationary kernels lead to models that underperform in practice. Shallow – or single – Gaussian processes are often suboptimal since flexible kernels that would account for the non-stationary and long-range connections of the data are difficult to design and infer. Deep Gaussian processes elevate the performance of Gaussian processes by mapping the inputs through multiple Gaussian process 'layers' [Damianou and Lawrence, 2013, Salimbeni and Deisenroth, 2017]. However, deep GP's result in degenerate models if the individual GP's are not invertible, which limits their capacity [Duvenaud et al., 2014].

In this abstract, we explore a novel paradigm of learning continuous-time transformations or *flows* of the data instead of learning a discrete sequence of layers. We apply GP based stochastic differential equation systems in the original data space to transform the inputs before a GP classification or regression layer. The transformation flow consists of an infinite path of infinitesimal steps. This approach turns the focus from learning iterative function mappings to learning input representations in the original feature space, avoiding learning new feature spaces.



(a) Deep Gaussian process



(b) Differentially deep Gaussian process

Figure 1: **(a)** Deep Gaussian process is a hierarchical model with a nested composition of Gaussian processes introducing *dependency across layers*. **(b)** In our formulation, deepness is introduced as a *temporal dependency across states* $\mathbf{x}_i(t)$ (indicated by dashed line) with a GP prior over the differential function $\mathbf{f}$

2 Model

We propose a *continuous-time* modeling approach where inputs $\mathbf{x}_i$ are not treated as constant but are instead driven by an SDE system. In particular, we consider process warping an input $\mathbf{x}$ through a differential function $\mathbf{f}$ until a predefined time T, resulting in $\mathbf{x}(T)$, which is subsequently classified or regressed with another function g. We impose GP priors on both the differential

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

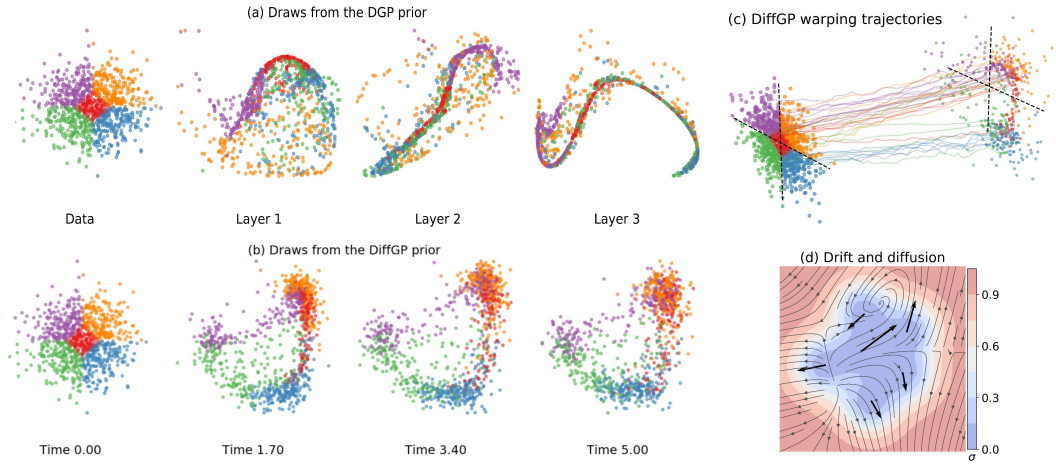


Figure 2: **(a)** Illustration of samples from a 2D deep Gaussian processes prior. DGP prior exhibits a pathology wherein representations in deeper layers concentrate on low-rank manifolds. **(b)** Samples from a differentially deep Gaussian processes prior result in rank-preserving representations. **(c)** The continuous-time nature of the warping trajectories results from smooth drift and structured diffusion **(d)**.

field $\mathbf{f}$ and the predictor function g. A key parameter of the differential GP model is the amount of simulation time T, which defines the length of flow and the capacity of the system, analogously to the number of layers in standard deep GPs. The framework reduces to a conventional Gaussian process with zero flow time $T = 0$.

We assume a dataset of N inputs $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T \in \mathbb{R}^{N \times D}$ of D-dimensional vectors $\mathbf{x}_i \in \mathbb{R}^D$, and associated scalar outputs $\mathbf{y} = (y_1, \ldots, y_N)^T \in \mathbb{R}^N$ that can be continuous for a regression problem or categorical for classification, respectively. We redefine the inputs as temporal functions $\mathbf{x} : \mathcal{T} \to \mathbb{R}^D$ over time such that state paths $\mathbf{x}_t$ over time $t \in \mathcal{T} = \mathbb{R}_+$ emerge, where the observed inputs $\mathbf{x}_{i,t} \triangleq \mathbf{x}_{i,0}$ correspond to initial states $\mathbf{x}_{i,0}$ at time 0. We classify or regress the final data points $\mathbf{X}_T = (\mathbf{x}_{1,T}, \ldots, \mathbf{x}_{N,T})^T$ after T time of an SDE flow with a predictor Gaussian process g.

In addition, we consider sparse Gaussian process approach by augmenting both differential and predictor Gaussian process with inducing variables with GP prior [Snelson and Ghahramani, 2006, Titsias, 2009]. The joint density of the augmented model is as below.

$$p(\mathbf{y}, \mathbf{g}, \mathbf{u}_g, \mathbf{X}_T, \mathbf{f}, \mathbf{U}_\mathbf{f}) = \underbrace{p(\mathbf{y}|\mathbf{g})}_{\text{likelihood}} \underbrace{p(\mathbf{g}|\mathbf{u}_g, \mathbf{X}_T)p(\mathbf{u}_g)}_{\text{GP prior of } g(\mathbf{x})} \underbrace{p(\mathbf{X}_T|\mathbf{f})}_{\text{SDE}} \underbrace{p(\mathbf{f}|\mathbf{U}_\mathbf{f})p(\mathbf{U}_\mathbf{f})}_{\text{GP prior of } \mathbf{f}(\mathbf{x})}, \tag{1}$$

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

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

$$p(\mathbf{f}|\mathbf{U}_\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{Q}_\mathbf{f}\text{vec}(\mathbf{U}_\mathbf{f}), \mathbf{K}_{\mathbf{xx}} - \mathbf{Q}_\mathbf{f}\mathbf{K}_{\mathbf{Z}_\mathbf{f}\mathbf{Z}_\mathbf{f}}\mathbf{Q}_\mathbf{f}^T), \tag{3}$$

$$p(\mathbf{U}_\mathbf{f}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{u}_{\mathbf{f}d}|\mathbf{0}, \mathbf{K}_{\mathbf{Z}_\mathbf{f}\mathbf{Z}_\mathbf{f}}), \tag{4}$$

where $\mathbf{Q}_g = \mathbf{K}_{\mathbf{X}_T\mathbf{Z}_g}\mathbf{K}^{-1}_{\mathbf{Z}_g\mathbf{Z}_g}$ and $\mathbf{Q}_\mathbf{f} = \mathbf{K}_{\mathbf{X}\mathbf{Z}_\mathbf{f}}\mathbf{K}^{-1}_{\mathbf{Z}_\mathbf{f}\mathbf{Z}_\mathbf{f}}$. The model prediction depends on the distribution of the final states $p(\mathbf{X}_T|\mathbf{f})$ determined by the SDE flow $d\mathbf{x}_t$ of the input data $\mathbf{X}$. We define the flow parameterized by inducing vectors $\mathbf{U}_\mathbf{f}$ defining the vector field direction at 'landmark' locations $\mathbf{Z}_\mathbf{f}$. The drift and diffusion at every point in the data space is then defined with smooth non-linear Gaussian processes interpolation given by $p(\mathbf{f}|\mathbf{U}_\mathbf{f})$ in (3).

$$d\mathbf{x}_t = \boldsymbol{\mu}(\mathbf{x}_t)dt + \sqrt{\boldsymbol{\Sigma}(\mathbf{x}_t)}dW_t \tag{5}$$

where, drift $\boldsymbol{\mu}(\mathbf{x}_t) = \mathbf{K}_{\mathbf{x}_t\mathbf{Z}_\mathbf{f}}\mathbf{K}^{-1}_{\mathbf{Z}_\mathbf{f}\mathbf{Z}_\mathbf{f}}\mathbf{U}_\mathbf{f}$ is a deterministic state evolution vector field, $\boldsymbol{\Sigma}(\mathbf{x}_t) = \mathbf{K}_{\mathbf{x}_t\mathbf{x}_t} - \mathbf{K}_{\mathbf{x}_t\mathbf{Z}_\mathbf{f}}\mathbf{K}^{-1}_{\mathbf{Z}_\mathbf{f}\mathbf{Z}_\mathbf{f}}\mathbf{K}_{\mathbf{Z}_\mathbf{f}\mathbf{x}_t}$ is the diffusion scaling matrix of the stochastic multivariate Wiener process $W_t \in \mathbb{R}^D$. A Wiener process has zero initial state $W_0 = \mathbf{0}$, and independent, Gaussian increments $W_{t+s} - W_t \sim \mathcal{N}(\mathbf{0}, sI_D)$ over time with standard deviation $\sqrt{s}I_D$.

Optimizing the marginal log likelihood involves integrating out the state distributions $p(\mathbf{X}_T|\mathbf{f})$ of the non-linear SDE system (5) without a closed form solution. Instead, we propose to follow the SVI framework as considered by Hensman et al. [2015] to find a lower bound on the model evidence. In particular, a variational lower bound for the evidence 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 with Gaussian approximations for the inducing posteriors $q(\mathbf{u}_g) = \mathcal{N}(\mathbf{u}_g|\mathbf{m}_g, \mathbf{S}_g)$ and $q(\mathbf{U}_\mathbf{f}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{u}_{\mathbf{f}d}|\mathbf{m}_{\mathbf{f}d}, \mathbf{S}_{\mathbf{f}d})$.

$$q(\mathbf{g}, \mathbf{u}_g, \mathbf{X}_T, \mathbf{f}, \mathbf{U}_\mathbf{f}) = p(\mathbf{g}|\mathbf{u}_g, \mathbf{X}_T)q(\mathbf{u}_g)p(\mathbf{X}_T|\mathbf{f})p(\mathbf{f}|\mathbf{U}_\mathbf{f})q(\mathbf{U}_\mathbf{f}) \tag{6}$$

Further, the inducing variables $\mathbf{u}_g$ and $\mathbf{U}_\mathbf{f}$ can be marginalized out from the above joint distribution, arriving at following variational distributions (we refer to Hensman et al. [2015] for further details)

$$q(\mathbf{g}|\mathbf{X}_T) = \mathcal{N}(\mathbf{g}|\mathbf{Q}_g\mathbf{m}_g, \mathbf{K}_{\mathbf{X}_T\mathbf{X}_T} + \mathbf{Q}_g(\mathbf{S}_g - \mathbf{K}_{\mathbf{Z}_g\mathbf{Z}_g}), \mathbf{Q}_g^T), \tag{7}$$

$$q(\mathbf{f}) = \mathcal{N}(\mathbf{f}| \underbrace{\mathbf{Q}_\mathbf{f}\mathrm{vec}(\mathbf{M}_\mathbf{f})}_{\boldsymbol{\mu}_q}, \underbrace{\mathbf{K}_{\mathbf{XX}} + \mathbf{Q}_\mathbf{f}(\mathbf{S}_\mathbf{f} - \mathbf{K}_{\mathbf{Z}_\mathbf{f}\mathbf{Z}_\mathbf{f}})\mathbf{Q}_\mathbf{f}^T}_{\boldsymbol{\Sigma}_q}). \tag{8}$$

We plug the derived variational posterior drift $\boldsymbol{\mu}_q$ and diffusion $\boldsymbol{\Sigma}_q$ estimates to the final *variational SDE flow* which conveniently encodes the variational approximation of the vector field $\mathbf{f}$.

$$d\mathbf{x}_t = \boldsymbol{\mu}_q(\mathbf{x}_t)dt + \sqrt{\boldsymbol{\Sigma}_q(\mathbf{x}_t)}dW_t \tag{9}$$

The evidence lower bound for our differential deep GP model can be written as

$$\log p(\mathbf{y}) \geq \sum_{i=1}^{N} \left\{ \frac{1}{S} \sum_{s=1}^{S} \underbrace{\mathbb{E}_{q(\mathbf{g}|\mathbf{x}_{i,T}^{(s)})} \log p(y_i|g_i)}_{\text{variational expected likelihood}} - \underbrace{\mathrm{KL}[q(\mathbf{u}_g)||p(\mathbf{u}_g)]}_{\text{prior divergence of } g(x)} - \underbrace{\mathrm{KL}[q(\mathbf{U}_\mathbf{f})||p(\mathbf{U}_\mathbf{f})]}_{\text{prior divergence of } \mathbf{f}(x)} \right\}, \tag{10}$$

which factorizes over both data and Monte Carlo samples of SDE paths $\mathbf{x}_{i,T}^{(s)} \sim p_T(\mathbf{x}; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q, \mathbf{x}_i)$ defined by the numerical solutions of the system (9). We use Euler-Maruyama solver [Higham, 2001] to generate the numerical solutions of the posterior SDE. Assuming a fixed time discretisation $t_1, \ldots, t_k$ with $\Delta t = t_k/K$ being the time window, the EM method at t_k is

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \boldsymbol{\mu}_q(\mathbf{x}_k)\Delta t + \sqrt{\boldsymbol{\Sigma}_q(\mathbf{x}_k)}\Delta W_k, \tag{11}$$

where $\Delta W_k = W_{k+1} - W_k \sim \mathcal{N}(\mathbf{0}, \Delta t I_D)$ with standard deviation $\sqrt{\Delta t}$. However, in practice, more efficient methods with high-order approximations can be used as well [Kloeden and Platen, 1992, Lamba et al., 2006]. All the model parameters which include variational parameters and kernel hyperparameters are learned jointly using the stochastic gradients of the lowerbound defined by (10).

3 Experiments

3.1 Step function estimation

We begin by highlighting how the DiffGP estimates a signal with multiple highly non-stationary step functions as shown in the figure 3. The DiffGP separates the regions around the step function such that the final regression function g with a standard stationary Gaussian kernel can fit the transformed data $\mathbf{X}(t)$. The model then has learned the non-stationarities of the system with uncertainty in the signals being modelled by the inherent uncertainties arising from the diffusion.

3.2 UCI regression benchmarks

We compare our model on 8 regression benchmarks with the previously reported results in [Salimbeni and Deisenroth, 2017]. On Boston, Concrete and Power datasets, where deep models show improvement over shallow models, our model outperforms previous best results of DGP. There is a small improvement by having a non-linear model on the Kin8mn dataset and our results match that of DGP. Energy and Wine are small datasets where single Gaussian Processes perform the best. As expected, both DiffGP and DGP recover the shallow model indicating no over-fitting.

4 Discussion

We have proposed a continuous-time deep Gaussian process model wherein input locations are warped through stochastic and smooth differential equations. The continuous-time deep model admits 'decision-making paths', where we can explicitly follow the transformation applied to a data point $\mathbf{x}_i$. Analyzing these paths could lead to a better interpretable model. However, modeling in the input space without intermediate low-dimensional latent representations presents scalability issues. We leave scaling the approach to high dimensions as future work, while we also intend to explore other inference methods such as SG-HMC [Chen et al., 2014] or Stein inference [Liu and Wang, 2016] in the future.

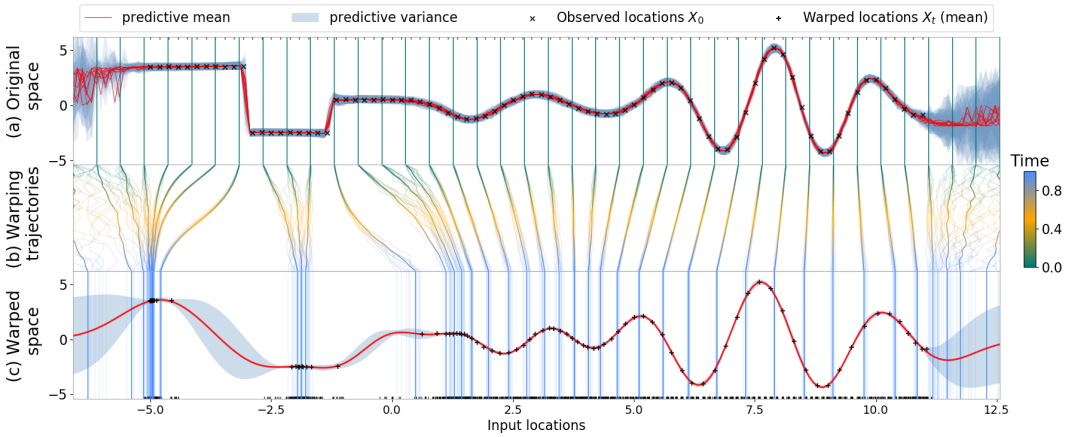


Figure 3: Step function estimation: Observed input space **(a)** is transformed through stochastic continuous-time mappings **(b)** into a warped space **(c)**. The stationary Gaussian process in the warped space gives a smooth predictive distribution corresponding to a highly non-stationary predictions in the original observed space.

		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 Salimbeni and Deisenroth [2017]). Uses random 90% / 10% training and test splits, repeated 20 times.

References

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International Conference on Machine Learning*, pages 1683–1691, 2014.

Andreas Damianou and Neil Lawrence. Deep gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215, 2013.

David Duvenaud, Oren Rippel, Ryan Adams, and Zoubin Ghahramani. Avoiding pathologies in very deep networks. In *Artificial Intelligence and Statistics*, pages 202–210, 2014.

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

Desmond Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Rev.*, 43:525–546, 2001.

P.E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Applications of Mathematics. Springer-Verlag, 1992. ISBN 9783540540625.

H Lamba, Jonathan C Mattingly, and Andrew M Stuart. An adaptive euler–maruyama scheme for sdes: convergence and stability. *IMA journal of numerical analysis*, 27:479–506, 2006.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in Neural Information Processing Systems*, pages 2378–2386, 2016.

C.E. Rasmussen and K.I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

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

Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, pages 1257–1264, 2006.

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