Transforming Gaussian Processes With Normalizing Flows

Juan Maroñas[*][†]
PRHLT Research Center
Universitat Politècnica
de València

Oliver Hamelijnck[†]
Dept. of CS
University of Warwick
The Alan Turing Institute

Jeremias Knoblauch
Dept. of Statistics
University of Warwick
The Alan Turing Institute

Theodoros Damoulas
Depts. of CS & Statistics
University of Warwick
The Alan Turing Institute

Abstract

Gaussian Processes (GPs) can be used as flexible, non-parametric function priors. Inspired by the growing body of work on Normalizing Flows, we enlarge this class of priors through a parametric invertible transformation that can be made input-dependent. Doing so also allows us to encode interpretable prior knowledge (e.g., boundedness constraints). We derive a variational approximation to the resulting Bayesian inference problem, which is as fast as stochastic variational GP regression (Hensman et al., 2013; Dezfouli and Bonilla, 2015). This makes the model a computationally efficient alternative to other hierarchical extensions of GP priors (Lázaro-Gredilla, 2012; Damianou and Lawrence, 2013). The resulting algorithm's computational and inferential performance is excellent, and we demonstrate this on a range of data sets. For example, even with only 5 inducing points and an input-dependent flow, our method is consistently competitive with a standard sparse GP fitted using 100 inducing points.

1 Introduction

Gaussian Processes (GPs) are perhaps the most well-known stochastic processes. Their popularity derives from their two most important features: not only are they infinite-dimensional generalizations of the multivariate normal distribution, but they also inherit numerous convenient properties from it. Most importantly, like its finite-dimensional counterpart, the GP is closed under marginalization and conditioning. Together, these features have made GPs uniquely attractive for modeling natural phenomena in molecular biology (Einstein, 1905), physics (Uhlenbeck and Ornstein, 1930), and spatial statistics (Krige, 1951).

Within Machine Learning, GPs are most commonly used as non-parametric Bayesian prior beliefs over functions, an idea dating back to O'Hagan (1978) and significantly expanded by Williams and Rasmussen (1996). Though GP priors can describe many functions, an ongoing line of work has constructed ever more expressive function priors at the expense of computational complexity (Snelson et al., 2003; Damianou and Lawrence, 2013; Wilson and Ghahramani, 2010; Lázaro-Gredilla, 2012; Garnelo et al., 2018). For instance, the work of Damianou and Lawrence (2013) and Lázaro-Gredilla (2012) considers layered compositions of GPs. While these priors are more expressive than single GPs, this comes at a price. For example, in the Deep GP (DGP) inference algorithm of Salimbeni and Deisenroth (2017), computations are $\mathcal{O}(NM^2 \cdot K + M^3 \cdot K)$, where N is the number of observations, $M << N$ the number of inducing points and K the total number of GPs (dozens per layer in the work of Salimbeni and Deisenroth, 2017).

The current paper produces a method capable of largely eliminating this trade-off between the expressivity and computational complexity of function priors: We present a simple yet powerful way of enlargening the class of GP priors without substantially increasing computational cost. Building on Wilson and Ghahramani (2010) and Wauthier and Jordan (2010), we apply parametric and invertible transformations (aka Normalizing Flows (Rezende and Mohamed, 2015)) to a GP—yielding a Transformed GP (TGP). In contrast to previous approaches however, the TGP also allows for Bayesian, input-dependent transformations. The TGP is more expressive than a GP, and can encode additional prior knowledge about the latent function (e.g., boundedness constraints). With a sparse variational inference scheme, the TGP's run time

[*] Work started while at an internship at the Alan Turing Institute. [†] Indicates equal contribution. Order decided using `numpy.random.randint(2)`. Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

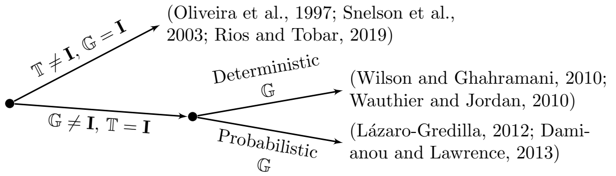
$\mathbb{T} \neq \mathbf{I}, \mathbb{G} = \mathbf{I}$
(Oliveira et al., 1997; Snelson et al., 2003; Rios and Tobar, 2019)
Deterministic $\mathbb{G}$
(Wilson and Ghahramani, 2010; Wauthier and Jordan, 2010)
$\mathbb{G} \neq \mathbf{I}, \mathbb{T} = \mathbf{I}$
Probabilistic $\mathbb{G}$
(Lázaro-Gredilla, 2012; Damianou and Lawrence, 2013)

Figure 1: A simplified categorisation of some of the literature on transforming GPs.

is $\mathcal{O}(NM^2 + M^3)$—virtually identical to that of standard sparse variational GP (SVGP) regression (Hensman et al., 2013). Further, the TGP outperforms the SVGP using only a fraction of its inducing points and produces test performances comparable to a multi-layer DGP.

While this is not a focus of the current paper, our inference scheme can also easily incorporate transformations of the data. This means we also provide a faster approximation to a number of previous models, including the Warped Gaussian Process (Snelson et al., 2003; Rios and Tobar, 2019).

2 Motivation

Existing literature uses invertible transformations within GPs either on the prior or the likelihood. For observations $(\mathbf{X}, \mathbf{Y})$ and invertible mappings $\mathbb{G}, \mathbb{T}$, a generative model unifying both approaches is

$$\left.\begin{array}{l} \mathbf{f}_0 \sim \text{GP}(\mu(\cdot), C(\cdot, \mathbf{X}));\ \ \mathbf{f}_K = \mathbb{G}(\mathbf{f}_0) \\ \mathbb{T}(\mathbf{Y}) = \mathbf{f}_K + \epsilon;\ \ \epsilon \overset{iid}{\sim} \mathcal{N}(0, \Sigma). \end{array}\right\} \tag{1}$$

Denoting $\mathbf{I}$ as the identity function, this recovers standard GP regression for $\mathbb{G} = \mathbb{T} = \mathbf{I}$. Similarly, setting $\mathbb{T} \neq \mathbf{I}, \mathbb{G} = \mathbf{I}$ / $\mathbb{G} \neq \mathbf{I}, \mathbb{T} = \mathbf{I}$ amounts to transforming only the likelihood / prior. Clearly, it is also possible to incorporate additional Bayesian priors about $\mathbb{G}$ and $\mathbb{T}$. In this case, the transformation itself becomes probabilistic. Fig. 1 illustrates this categorization.

2.1 Related work

The arguable more popular subcase of Eq. (1) is applied to the likelihood ($\mathbb{T} \neq \mathbf{I}$, $\mathbb{G} = \mathbf{I}$). The methods resulting from this strategy are well-studied in spatial statistics, and also known as Trans-kriging models (Diggle and Ribeiro, 2007). The earliest work is based on exponential transforms such as the Box-Cox (Box and Cox, 1964), but transformations soon took various other forms, including the Tukey transform (Tukey, 1977), hyperbolic transformations (Tsai et al., 2017), and the Sinh-Archsinh transforms (Jones and Pewsey, 2009). Transformation parameters can be estimated (Snelson et al., 2003) or integrated out via Bayes' rule (Oliveira et al., 1997; Muré, 2018). In essence, transforming the likelihood is an attempt at Gaussianization (Chen and Gopinath, 2000; Meng et al., 2020): One hopes that $\mathbb{T}$ makes $\mathbb{T}(\mathbf{Y}) = \boldsymbol{Z}$ into a standard GP with additive noise (Lin and Joseph, 2019). Note that whenever $\mathbb{T}$ is non-linear, this implies that $\mathbf{Y} = \mathbb{T}^{-1}(\boldsymbol{Z})$ is non-Gaussian with non-additive noise. Thus, one can also see these transformations as aiming to fix model misspecification. This means that unlike other approaches towards robustifying GPs (see e.g. Jylänki et al., 2011; Hartmann and Vanhatalo, 2019; Knoblauch, 2019), transformations of the likelihood make sense *only* if one has sufficient domain knowledge to locate the source of misspecification.

On the other hand, transformations of the prior ($\mathbb{G} \neq \mathbf{I}, \mathbb{T} = \mathbf{I}$) have *no* implications for the likelihood model or error structure. Further—and unlike transformations of the likelihoods—they are applicable for discrete-valued data, too. This makes them more compatible with black box models —and so more attractive to the Machine Learning community. This does not mean that $\mathbb{G}$ cannot incorporate domain knowledge however—and we exploit this on two applications where we force the function prior to be non-negative.

These advantages have made the literature on transforming the GP prior an active research area. Some of its most prolific outcomes include Wilson and Ghahramani (2010) as well as the work of Wauthier and Jordan (2010) and Murray et al. (2009). In all three papers, the parameterizations of the transforms are deterministic. More recently, this was superseded by a probabilistic treatment (e.g. Monterrubio-Gómez et al., 2020). Deep GPs (Damianou and Lawrence, 2013) and Warped GPs (Lázaro-Gredilla, 2012) are perhaps the most prominent examples, and transform a base GP with a layered hierarchy of other GPs. A different line of work transforms the prior via the input $\mathbf{X}$ (see Calandra et al., 2016; Wilson et al., 2016), which induces non-stationarity relative to the original observation space without affecting the conditional Gaussianity of $\mathbf{Y}$.

2.2 Computation

Since the posterior of a standard GP regression ($\mathbb{G} = \mathbb{T} = \mathbf{I}$) has closed form, it is important to determine how much $\mathbb{G} \neq \mathbf{I}$ or $\mathbb{T} \neq \mathbf{I}$ complicates computations.

When the likelihood is transformed ($\mathbb{T} \neq \mathbf{I}$, $\mathbb{G} = \mathbf{I}$), marginal likelihoods often have closed forms (see e.g. Snelson et al., 2003). However, predictions need an explicit computation of the inverse $\mathbb{T}^{-1}$.

This leaves two options, both with considerable drawbacks: One can use approximation algorithms (e.g., Newton-Raphson) to approximate $\mathbb{T}^{-1}$, or one can constrain $\mathbb{T}$ to produce closed forms (see e.g. Rios and Tobar, 2019). The former bloats the computation and is sensitive to initial conditions, the latter constrains the model's flexibility.

When the prior is transformed ($\mathbb{T} = \mathbf{I}$, $\mathbb{G} \neq \mathbf{I}$), the inverse $\mathbb{G}^{-1}$ does *not* have to be computed explicitly. Such methods pose other challenges however: Their marginal likelihoods will not have closed form. In prior work, this has often resulted in rather coarse approximate inference. For example, Wilson and Ghahramani (2010) and Wauthier and Jordan (2010) set $\mathbb{G} \neq \mathbf{I}$ to obtain non-Gaussian marginals, but are forced to use Laplace approximations for inference.

The problem is compounded if $\mathbb{G}$ itself is probabilistic, as is the case for Deep GPs (DGPs). To address this issue, sparse GP priors and Variational Inference (VI) are typically used. For instance, Damianou and Lawrence (2013); Lázaro-Gredilla (2012) use a mean-field normal family. This is extended by Salimbeni and Deisenroth (2017) to capture uncertainty across layers. Unfortunately, both approximations have drawbacks, leading to recent work advocating for structured variational families instead (Ustyuzhaninov et al., 2020). This appears to produce better inferences, but also significantly increase the computational overhead.

2.3 Our Contribution

We design a Bayesian method that can match the performance of Deep GPs at a fraction of the computational cost. To achieve this, we consider Bayesian Neural Networks (NNs) as input-dependent parametric transforms. We then derive a sparse variational approximation extending the ideas of Titsias (2009); Hensman et al. (2013) and Dezfouli and Bonilla (2015).

Our approximation is also the first scalable variational method for the methods of Wilson and Ghahramani (2010); Wauthier and Jordan (2010). Further, our inference algorithm is applicable even if one also transforms the likelihood ($\mathbb{G} \neq \mathbf{I}, \mathbb{T} \neq \mathbf{I}$). This means that it can easily be adapted to incorporate domain knowledge via $\mathbb{G}$ (and $\mathbb{T}$), and we show this on two examples.

3 Model Description

Given N input-output tuples $\{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})\}_{i=1}^{N}$, we arrange them into matrices $\mathbf{X} \in \mathbb{R}^{N \times D_x}$ and $\mathbf{Y} \in \mathbb{R}^{N \times D_y}$. Throughout, our goal is the Bayesian learning over a set of functions $\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y}\}$. To this end, we place a prior distribution over a subset of possible functions $\mathcal{F}$. Given a GP specified via its mean and covariance functions $\mu(\cdot), C_\nu(\cdot, \cdot)$, we achieve this by additionally transforming it with K invertible parametric transformations $\{\mathbb{G}_{\theta_k}\}_{k=0}^{K-1}$. More precisely, we define for all $k = 0, \dots K-1$ the functions $\mathbb{G}_{\theta_k} : \mathcal{F} \to \mathcal{F}$ as the individual transformations, $\mathbb{G}_{\boldsymbol{\theta}} = \mathbb{G}_{\theta_0} \circ \mathbb{G}_{\theta_1} \circ \cdots \circ \mathbb{G}_{\theta_{K-1}}$ as their composition and $\boldsymbol{\theta} = \{\theta_0, \theta_1, \dots, \theta_{K-1}\}$ as the parameterization of this composition. Transformations of this kind have recently been popularized in a different context as flows (Rezende and Mohamed, 2015). While our model applies for $\boldsymbol{\theta} \in \mathbb{R}^p$, it also accommodates the case of function-valued (i.e. input-dependent) parameters $\boldsymbol{\theta} : \mathcal{X} \to \mathbb{R}^p$ such as Neural Networks (NNs).

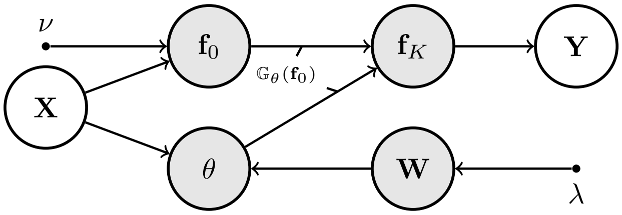

Figure 2: Plate diagram of the TGP with Bayesian input-dependent flows. We use transformations $\mathbb{G}_{\boldsymbol{\theta}}$ with input-dependent (function-valued) parameters $\boldsymbol{\theta} = \boldsymbol{\theta}(\mathbf{X}, \mathbf{W})$ to transform a base GP $\mathbf{f}_0$ with kernel hyperparameters ν into a more expressive prior $\mathbf{f}_K$ about the functional relationship between $\mathbf{X}$ and $\mathbf{Y}$. In practice, $\mathbf{W}$ will be weights of a Neural Network (NN) with a Bayesian prior depending on hyperparameters λ.

3.1 The Transformed Gaussian Process (TGP)

Taking $\mathbf{f}_0 \sim \text{GP}(\mu(\mathbf{X}), C_\nu(\mathbf{X}, \cdot))$ as a sample from the base GP, we then define the TGP as $\mathbf{f}_K = \mathbb{G}_{\boldsymbol{\theta}}(\mathbf{f}_0)$. For simplicity, the current paper restricts attention to element-wise mappings. Because such mappings produce diagonal Jacobians, they only affect the marginals of the GP, so that for any fixed $\mathbf{X}' \in \mathcal{X}$, $\mathbf{f}_K(\mathbf{X}') = \mathbb{G}_{\boldsymbol{\theta}}(\mathbf{f}_0(\mathbf{X}'))$. Thus, we will often refer to them as *diagonal/marginal* transformations/flows. Note that the resulting TGP $\mathbf{f}_K$ can be seen as an input-dependent generalization of the Gaussian Copula Process discussed in Wilson and Ghahramani (2010).

3.2 Input-dependent Flows

A simple example for a marginal flow is given by stacking K SAL flows (Rios and Tobar, 2019):

$$\left.\begin{aligned}\mathbf{f}_1 &= d_1 \cdot \sinh(b_1 \cdot \operatorname{arcsinh}(\mathbf{f}_0) - a_1) + c_1 \\ &\dots \\ \mathbf{f}_K &= d_K \cdot \sinh(b_K \cdot \operatorname{arcsinh}(\mathbf{f}_{K-1}) - a_K) + c_K\end{aligned}\right\} \quad (2)$$

In this example, $\mathbb{G}_{\boldsymbol{\theta}}$ is not input-dependent and $\theta_{j-1} = \{a_j, b_j, c_j, d_j\}$. Fig. 3 illustrates the effect of such a transform on a base GP for $K = 3$. We could make the transformation input-dependent however, the only thing required is a reparameterization. In particular, one only has to replace the scalar parameters a_j, b_j, c_j, and d_j with the function-valued parameters $\alpha_j, \beta_j, \gamma_j, \delta_j : \mathcal{X} \to \mathbb{R}$.

We achieve this via Neural Networks (NNs) with L layers so that for any fixed $\mathbf{X}' \in \mathcal{X}$, the transformation's parameters are $\{\alpha_j(\mathbf{X}'), \beta_j(\mathbf{X}'), \gamma_j(\mathbf{X}'), \delta_j(\mathbf{X}')\}_{j=0}^{K-1}$. Thus, if the NN's weights $\{\mathbf{W}^l\}_{l=1}^{L}$ are fitted without accounting for parameter uncertainty, $\boldsymbol{\theta} = \{\mathbf{W}^l\}_{l=1}^{L}$. Note that a model of this form will be able to model non-stationary processes. We illustrate this using a range of warping functions at different locations in App. B.7.2.

3.3 Bayesian Priors on Flows

However, we find that a Bayesian treatment of $\{\mathbf{W}^l\}_{l=1}^{L}$ significantly improves test set performance. This is hardly surprising; input-dependent flows in the form of NNs introduce a considerable number of additional hyperparameters, making a naive implementation prone to over-fitting. The reason for this is that enriching GP priors with non-Bayesian flows provides additional fexibility via hyperparameters which are not regularized via a complexity penalty at inference time. By placing a Bayesian prior $p(\mathbf{W})$ on the network weights $\mathbf{W} = \{\mathbf{W}^l\}_{l=1}^{L}$, we effectively regularize the network weights and avoid this issue. This means that we integrate over $\{\mathbf{W}^l\}_{l=1}^{L}$, accounting for uncertainty in $\boldsymbol{\theta}$.

Though the prior could be chosen arbitrarily, we consider the fully factorized normal prior $p_\lambda(\mathbf{W}) = \mathcal{N}(\mathbf{W}; 0, \lambda^{-1} I_{|\mathbf{W}|\times|\mathbf{W}|})$ throughout the paper. The corresponding graphical model is given in Fig. 2, and the generative process is

$$\mathbf{f}_0|\mathbf{X} \sim \text{GP}(\mu(\mathbf{X}), C_\nu(\mathbf{X}, \cdot)) \qquad \mathbf{W} \sim p_\lambda(\mathbf{W})$$
$$\boldsymbol{\theta}(\mathbf{X}, \mathbf{W}) = \text{NN}(\mathbf{X}, \mathbf{W}) \qquad \mathbf{f}_K|\boldsymbol{\theta}, \mathbf{X}, \mathbf{W} = \mathbb{G}_{\boldsymbol{\theta}(\mathbf{X},\mathbf{W})}(\mathbf{f}_0)$$

Unlike in previous work (e.g. Wilson and Ghahramani, 2010), we not only quantify uncertainty about the parameter $\boldsymbol{\theta}$, but also make it an input-dependent function.

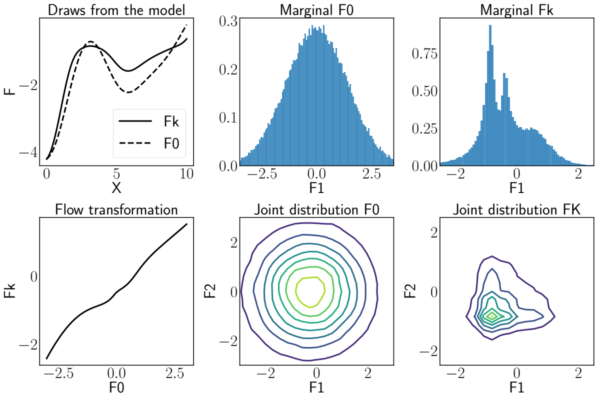

Figure 3: Flow constructed as in Eq. (2) with $K = 3$. The parameters of the flow were obtained from one of the experiments run in this work.

3.4 Induced Distributions

By virtue of an iterated application of the change of variable formula and the inverse function theorem (see e.g. Rezende and Mohamed, 2015), the probability distributions induced by our transformations are:

$$p(\mathbf{f}_K|\mathbb{G}, \mathbf{X}) = p(\mathbf{f}_0|\mathbf{X}) \prod_{k=0}^{K-1} \left| \det \frac{\partial \mathbb{G}_{\theta_k}(\mathbf{f}_k)}{\partial \mathbf{f}_k} \right|^{-1}.$$

By using a marginal flow, Sklar's theorem (Sklar, 1959) implies that the dependencies in $p(\mathbf{f}_0)$ and $p(\mathbf{f}_K)$ are driven by the same Copula—the GP in our case. Though the copula is the same, $\mathbf{f}_K$ will generally have non-Gaussian marginals (see Fig. 3). While the current paper restricts attention to diagonal mappings for simplicity, the presented derivations and methods may be extended to non-diagonal transformations such as those in Rios (2020). In practical terms, non-diagonal transformations could be used to model arbitrary copulas and correlation structures; and we elaborate on this version of the model in App. A. Whether $\mathbb{G}_{\boldsymbol{\theta}}$ is a diagonal or non-diagonal transformation, we require that the resulting $\mathbf{f}_K$ is a valid stochastic process (and thus a valid function prior). This amounts to checking whether the resulting collection of random variables satisfies the necessary consistency conditions, which holds by simple arguments for marginal flows (e.g. Rios, 2020). In order to employ flows such as Real NVP (Dinh et al., 2017), one needs to prove that these conditions are still satisfied. We leave this for future work, as the associated theory is highly dependent on the exact flow in question.

4 Inference

Performing inference for the Transformed Gaussian Process (TGP) is generally intractable. Thus, we derive an efficient sparse variational approximation which is amenable to stochastic optimization (Hensman et al., 2013), utilizes inducing points through sparse GP priors (Titsias, 2009) and works with arbitrary likelihoods

(Hensman et al., 2015; Dezfouli and Bonilla, 2015). An important part of this is a careful choice of the variational posterior, which eliminates the need to compute Jacobians or the inverse forms of $\mathbb{G}_{\boldsymbol{\theta}}$ altogether, yielding a drastic speedup. As a result, our inference algorithm is of order $\mathcal{O}(NM^2 + M^3)$ (for $M << N$) and parameters can be set via stochastic optimization. This makes our method particularly suitable for large-scale applications—and much faster than previous approaches: for example, the case of deterministic parametric transformations presented by Wauthier and Jordan (2010) and Wilson and Ghahramani (2010) relies on $\mathcal{O}(N^3)$ Laplace approximations. Similarly, the approximations for the case of hierarchical probabilistic transformations presented in Salimbeni and Deisenroth (2017) are of order $\mathcal{O}(NM^2 \cdot K + M^3 \cdot K)$, where K is the number of GPs inside the Deep GP (dozens per layer in the work of Salimbeni and Deisenroth, 2017).

4.1 Sparse Variational Objective

Variational methods frame inference as optimization by minimizing the Kullback-Leibler divergence (KL) between approximate and true posterior (Blei et al., 2017). It can also be interpreted as constrained finite-dimensional version of the infinite-dimensional variational problem characterizing the exact Bayesian posterior (Knoblauch et al., 2019). Rewriting this minimization as maximization, it becomes an Evidence Lower Bound (ELBO) (Bishop, 2006).

Sparse GPs augment the prior with M inducing points $\mathbf{u}_0 \in \mathbb{R}^M$ at locations $\mathbf{Z} \in \mathbb{R}^{M\times D}$, typically with $M < N$. These points act as 'pseudo-observations' and allow low rank approximations to the GP prior that circumvent the cubic costs traditionally associated with GP inference (Quiñonero Candela and Rasmussen, 2005; Williams and Seeger, 2001).

Taking the inducing points into account, the sparsified and transformed prior of the TGP is given by

$$p(\mathbf{f}_K, \mathbf{u}_K) = \underbrace{p(\mathbf{f}_0 \mid \mathbf{u}_0)\mathbf{J}_{\mathbf{f}_K}}_{=p(\mathbf{f}_K|\mathbf{u}_K)} \underbrace{p(\mathbf{u}_0)\mathbf{J}_{\mathbf{u}_K}}_{=p(\mathbf{u}_K)}, \tag{3}$$

where $p(\mathbf{u}_0)$ is a GP prior of the same form as that for $\mathbf{f}_0$ in Eq. (1) , $p(\mathbf{f}_0 \mid \mathbf{u}_0)$ is conditionally Gaussian and $\mathbf{J}_{\mathbf{a}} = \prod_{k=0}^{K-1} \left|\det \frac{\partial \mathbb{G}_{\theta_k}(\mathbf{a})}{\mathbf{a}}\right|^{-1}$ is the (diagonal) Jacobian of the transformation of the stochastic process $\mathbf{a}$. Rios (2020) formally proves that stochastic processes transformed by such marginal flows induce valid stochastic processes. In turn, this guarantees that the transformed sparse stochastic process is consistent—and thus a valid function prior.

One important property of the original bound proposed by Titsias (2009) is that the variational posterior implicitly cancels the conditional, as this alleviates the need for computing $\mathcal{O}(N^3)$ matrix inverses (Bauer et al., 2016). Using the same algebraic tricks, we define our approximate posterior such that *both* the conditional and the Jacobians cancel

$$q(\mathbf{f}_K, \mathbf{u}_K) = p(\mathbf{f}_K|\mathbf{u}_K)\underbrace{q(\mathbf{u}_0)\mathbf{J}_{\mathbf{u}_K}}_{=q(\mathbf{u}_K)}$$

where the $p(\mathbf{f}_K|\mathbf{u}_K)$ and $\mathbf{J}_{\mathbf{u}_K}$ are defined as before and $q(\mathbf{u}_0) = \mathcal{N}(\mathbf{u}_0 \mid \mathbf{m}, \mathbf{S})$ is a free form Gaussian with $\mathbf{m} \in \mathbb{R}^{M\times 1}$ and $\mathbf{S} \in \mathbb{R}^{M\times M}$. Another crucial side-effect of defining the variational posterior in this way is that it allows us to integrate out $\mathbf{u}_K$ analytically. Following Hensman et al. (2013) we do not collapse $q(\mathbf{u}_0)$ in order for stochastic Variational Inference to scale, resulting in the following ELBO:

$$\mathcal{L}(\mathbf{Y}) = \mathbb{E}_{q(\mathbf{f}_K,\mathbf{u}_K)}\left[\log \frac{p(\mathbf{Y} \mid \mathbf{f}_K)\cancel{p(\mathbf{f}_K \mid \mathbf{u}_K)}p(\mathbf{u}_0)\cancel{\mathbf{J}_{\mathbf{u}_K}}}{\cancel{p(\mathbf{f}_K \mid \mathbf{u}_K)}q(\mathbf{u}_0)\cancel{\mathbf{J}_{\mathbf{u}_K}}}\right]$$

which after some algebraic manipulations (see App. A) simplifies to our proposed variational lower bound:

$$\sum_{n=1}^{N} \mathbb{E}_{q(\mathbf{f}_{0,n})}\left[\log p(\mathbf{Y}_n \mid \mathbb{G}_{\boldsymbol{\theta}}(\mathbf{f}_{0,n}))\right] - \mathrm{KL}\left[q(\mathbf{u}_0) \mid\mid p(\mathbf{u}_0)\right].$$

The use of marginal flows and factorizing likelihoods results in the expected log likelihood (ELL) term being decomposable across the latent variables $q(\mathbf{f}_{0,n})$ and observations $\mathbf{Y}_n$—making it particularly suitable for stochastic variational inference and big N (Hensman et al., 2013). The individual ELL components will generally be unavailable in closed form and computed using one-dimensional Gaussian quadrature (Hensman et al., 2015).

4.2 Justification of Approximate Family

To understand the effect of our proposed approximate family we now compare against the case where the variational posterior is constrained to be Gaussian. We denote this model as G-SP and provide the full derivation in App. A.5.1. Because only the prior is transformed, the sparse conditional $p(\mathbf{f}_K \mid \mathbf{u}_K)$ no longer cancels; leading to an ELBO requiring $\mathcal{O}(N^3)$ computations and thus limiting G-SP to only small scale experiments. Additionally, G-SP requires passing samples of the Gaussian approximate posterior through the inverse prior transformation which can be undefined, for example, with positive enforcing flows. For G-SP, we thus only consider flows that leave the output space unconstrained.

As demonstrated in Fig. 4 the TGP is able to achieve a superior performance by fitting the flat regions of the data. The G-SP, although enriched with a flow on the prior, cannot learn more complex, space constraining flows, and instead learns a transformation that is close to identity and hence almost recovers the SVGP. In comparison TGP enriches both the prior and the approximate posterior family, increasing the flexibility of both whilst achieving superior performance and maintaining the computation benefits of the SVGP.

4.3 A Sparsification of Previous Models

Note that our approximation directly provides a new sparse variational inference algorithm for the models proposed by Wilson and Ghahramani (2010) and Wauthier and Jordan (2010). In methodological terms, it is a direct generalization of Hensman et al. (2013) (for $\mathbb{G}_{\boldsymbol{\theta}} \neq \mathbf{I}$). Crucially, this means that while our model allows for substantially more expressive function priors, it inherits the computational efficiency of the standard variational GP model.

While we focus mainly on prior transformations, our variational approximation also allows for transformations of the likelihood. Following Snelson et al. (2003) the transformed likelihood is given by:

$$p(\mathbf{Y} \mid \mathbb{T}, \mathbf{f}_K) = p(\mathbb{T}(\mathbf{Y}) \mid \mathbf{f}_K) \underbrace{\prod_{k=0}^{K-1} \left| \det \frac{\mathbb{T}_k(\mathbf{Y}_k)}{\mathbf{Y}_k} \right|}_{\mathbf{J}_{\mathbf{Y}_K}}.$$

where the Jacobian term does not depend on $\mathbf{f}_0$. Substituting this into our bound results in

$$\bar{\mathcal{L}} = \mathcal{L}(\mathbb{T}(\mathbf{Y})) + \log \mathbf{J}_{\mathbf{Y}_K}.$$

Setting $\mathbb{G}_{\boldsymbol{\theta}} = \mathbf{I}$, this constitutes the first sparse variational approximation to the work of Snelson et al. (2003) and Rios and Tobar (2019). Unlike previous inference schemes available for these models, this makes them applicable to large-scale applications through sparse GPs and mini-batching. We demonstrate this contribution in our experiments, and call this new inference scheme variational Warped GP (V-WGP).

4.4 Bayesian Input-dependent Flows

For input-dependent flows whose parameters have Bayesian priors, we specify priors and variational posteriors that are independent of $\mathbf{f}_K$. For the case of the input-dependent flow being a NN, this means that

$$p(\mathbf{f}_K, \mathbf{W}) = p(\mathbf{f}_K) p_\lambda(\mathbf{W})$$
$$q(\mathbf{f}_K, \mathbf{W}) = q(\mathbf{f}_K) q(\mathbf{W})$$

Absorbing these additional terms into the ELBO adds $-\mathrm{KL}[q(\mathbf{W}) || p(\mathbf{W})]$ to our bound. Additionally, the ELL term now requires integration over $q(\mathbf{W})$. The resulting integral could be approximated with S Monte Carlo samples $\{W_s\}_{s=1}^S$ using reparameterization, allowing unbiased gradient estimates of low variance to be computed via the approximation

$$\sum_{n=1}^{N} \mathbb{E}_{q(\mathbf{W})q(\mathbf{f}_{0,n})} \left[\log p(\mathbf{Y}_n \mid \mathbb{G}_{\boldsymbol{\theta}}(\mathbf{f}_{0,n}))\right] \approx$$
$$\sum_{n=1}^{N} \frac{1}{S} \sum_{s=1}^{S} \mathbb{E}_{q(\mathbf{f}_{0,n})} \left[\log p(\mathbf{Y}_n \mid \mathbb{G}_{\boldsymbol{\theta}(\mathbf{X},\mathbf{W}_s)}(\mathbf{f}_{0,n}))\right];$$

In our experiments however, we instead use the Monte Carlo Dropout approximation (Gal and Ghahramani, 2016) as it is more memory-efficient and avoids well-known pathologies introduced by mean field approximate posteriors $q(\mathbf{W})$ (e.g. Turner and Sahani, 2011). To study the effect this has relative to standard Variational Bayes, we provide a comparative study in App. B.6.3.

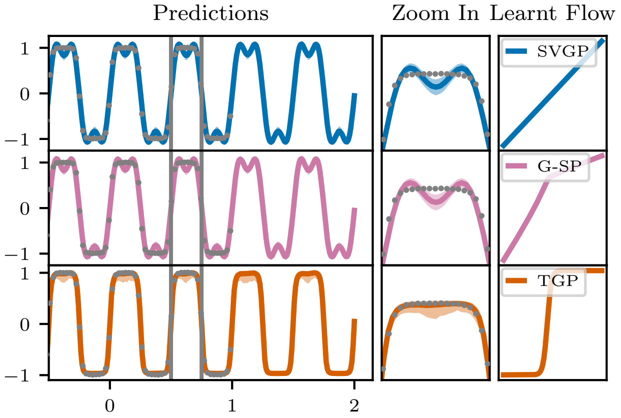

Figure 4: We generate 150 observations by passing evenly spaced evaluations of a sine curve, through a tanh flow and additionally applying Gaussian noise. For TGP we use a tanh flow and for G-SP we use a SAL, because G-SP requires a flow that does not constrain the output space. For all models we use a periodic kernel and perform a grid search across and a period initial value of $[0.6, 0.7, 0.8, 0.9, 1.0]$ and flow initialisation of identity, random and from data. We plot the best performing result from each model. The SVGP is unable to accurately model the flat regions of the dataset, whereas TGP can due to the prior transformation. G-SP uses a less suitable flow and so almost recovers the SVGP.

4.5 Prediction

To predict using the TGP ($\mathbb{G}_{\boldsymbol{\theta}} \neq \mathbf{I}, \mathbb{T} = \mathbf{I}$) we substitute the true posterior for its approximation $q(\mathbf{f}_K)$. The predictive distribution is then given by

$$p(\mathbf{Y}^*) = \int p(\mathbf{Y}^* \mid \mathbb{G}_{\boldsymbol{\theta}}(\mathbf{f}_0)) q(\mathbf{f}_0) \, \mathrm{d}\mathbf{f}_0 \,.$$

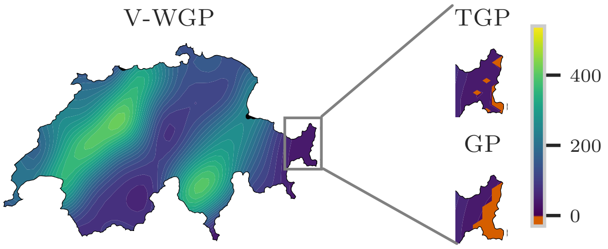

Figure 5: Spatial median predictions from a V-WGP, TGP and GP on the Switzerland daily rainfall dataset (units in 10 μm). The V-WGP (left) is guaranteed to have non-negative predictions. The TGP and GP (right) do not and predict negative rainfall in Graubünden.

For predictions, we use one-dimensional quadrature to approximate the first moment of $p(\mathbf{Y}^*)$. Confidence intervals are obtained by sampling from $p(\mathbf{Y}^*)$. Further details on predicting with Bayesian input-dependent flows and transformed likelihoods are in App. A.

5 Experiments

We first compare the TGP with likelihood-transforming methods. Second, we demonstrate the TGP's excellent performance as a black box model by using input-dependent Bayesian flows on a range of UCI datasets (Lichman, 2013). All code is written in PyTorch (Paszke et al., 2019) using GPyTorch (Gardner et al., 2018). Details can be found in App. B, and the code is publicly available at `https://github.com/jmaronas/TGP.pytorch`.

For all figures, we use the following acronyms: TGP (non input-dependent TGP), BA-TGP (input-dependent TGP with Bayesian flows) and PE-TGP (input-dependent TGP whose flow parameterization is obtained through a point estimate).

5.1 Applications

We first study two applications to compare transformations of priors with those of likelihoods. We restrict the TGP to non-input-dependent flows and we compare it against a scalable variational approximation for the Warped GP model of Snelson et al. (2003) (henceforth V-WGP) that we derive in Sec. 4.3 and the Sparse Variational GP (SVGP) of Hensman et al. (2013).

Air Quality Consider Particulate Matter of 2.5 μm size (PM25) in London (London, 2020) as depicted in Fig. 6. Measurements of PM25 are non-negative, exhibit periodic fluctuations due to vehicle traffic, and irregular peaks arising from weather conditions or traffic jams. Thus, we choose a SAL and softplus composition (SAL+SP). This makes the TGP's latent function positive and guarantees $\mathbb{T}(\mathbf{Y}) \geq 0$ for the V-WGP.

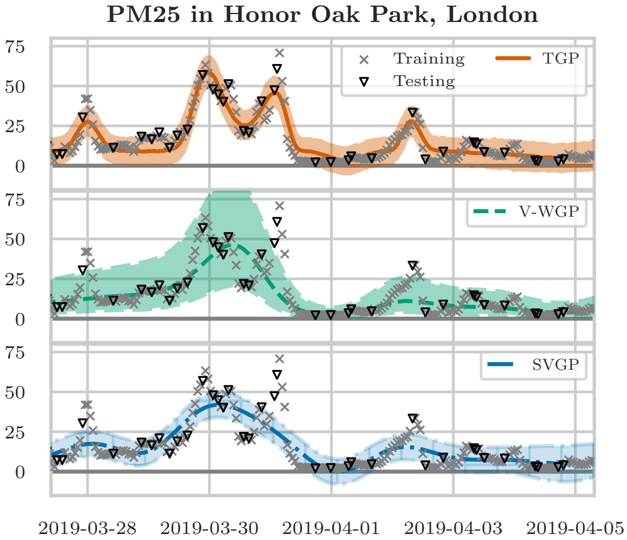

Figure 6: Model fits on PM25 with 5% of inducing points. The TGP's added flexibility provides the best fit. **Top**: the TGP with compositional SAL and softplus flow. **Middle**: V-WGP with the same flow, but in reverse and applied to the likelihood. **Bottom**: SVGP.

The difference between methods is noticeable for low numbers of inducing points (see Fig. 6 & 7). As discussed in Sec. 2, the V-WGP implicitly models $\mathbf{Y}$ with non-additive noise while the TGP transforms the prior, but models the noise additively. Hence, the TGP will attribute fluctuations to the underlying latent function, while the V-WGP is prone to absorb oscillations into the observation noise, as in Fig. 6. Unsurprisingly, the TGP's fit is superior to that of the GP due to its additional flexibility. However, even though $\mathbb{G}_{\boldsymbol{\theta}}$ is chosen so that $\mathbb{G}_{\boldsymbol{\theta}}(\mathbf{f}_0) \geq 0$, the TGP assigns positive probability mass to PM25 being negative.

Switzerland Rainfall We also model daily rainfall in Switzerland (Dubois, 2003), see Fig. 5. As observations are non-negative, we again employ SAL+SP flows. Unlike the TGP and GP, the V-WGP does not fit the latent function to peaks in the data and guarantees positive predictions. The resulting smoother fit is desirable and explains why the V-WGP's predictive performance in Fig. 7 outperforms that of the TGP.

5.2 Black Box results

We also highlight our model's capability to learn arbitrary functions in a Bayesian way on a range of regression and classification problems. Throughout, the TGP uses 1- or 2-layer NNs to parameterize input-dependent flows. While the inverses of these flows would be dif-

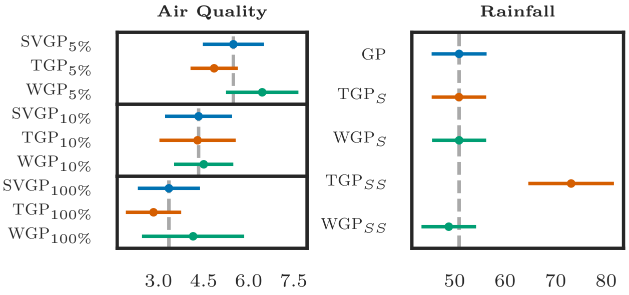

Figure 7: RMSE results (left is better) from the Air quality and Rainfall application. **Left**: Air quality experiments with 5%, 10% and 100% inducing points and a SAL plus softplus flow. The TGP consistently outperforms the GP and V-WGP because it can better fit to irregular patterns in the data. **Right**: Rainfall experiments with 100% inducing points and Softplus flows (S) versus SAL plus softplus flows (SS). When both the TGP and V-WGP use the more expressive SS flow, the V-WGP is superior, reflecting that the source of misspecification is the likelihood, not the prior.

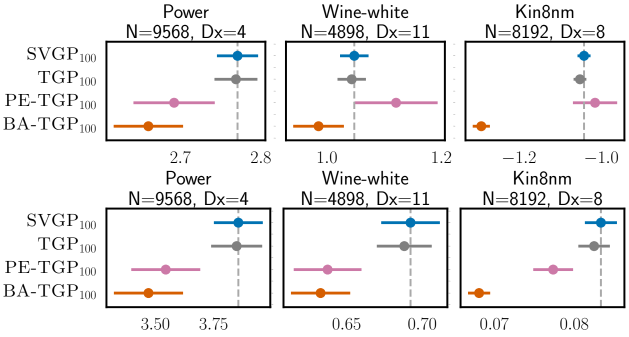

Figure 8: Comparison of NLL (top; left is better) and RMSE (bottom; left is better) for a standard SVGP with a non input-dependent flow (TGP), the input-dependent counterpart indexed by a NN when the NN is fitted using a point estimate (PE-TGP) or integrated out in a Bayesian fashion (BA-TGP)

ficult to approximate, inference for the TGP can proceed without computing the inverse transformations. This is a clear distinction to methods like the V-WGP, which relies on transforming the likelihood instead of the prior. We present results for the negative log likelihood (NLL) as they are representative for the overall findings and defer RMSE values, 95% COVERAGE and accuracy metrics to App. B together with details on choosing flows and NN architectures. For all plots, subscripts denote the number of inducing points used.

Bayesian Regularization First, we investigate the effect of Bayesian marginalization of the input-dependent flows by comparing RMSE and NLL. In particular, we use a NN to induce input-dependence for the flow and compare the results obtained by using a point estimate via standard dropout (Srivastava et al., 2014) versus those obtained with approximate Bayesian marginalization via Monte Carlo Dropout (Gal and Ghahramani, 2016). The results are depicted in Fig. 8 and demonstrate that preventing the NN from overfitting with a Bayesian treatment yields a significant performance boost in terms of predictive uncertainty. We also illustrate how non input-dependent flows are much less expressive than the input-dependent counterpart.

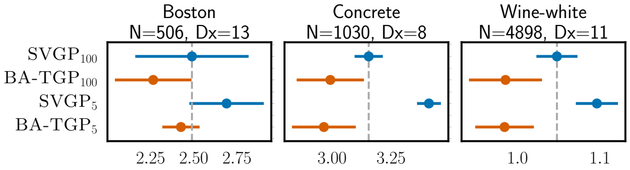

Figure 9: Comparing NLL (left is better) for some medium-sized regressions with 5 and 100 inducing points. Remaining data sets in App. B.

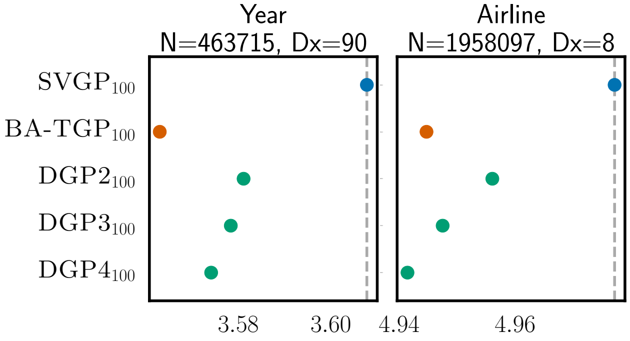

Figure 10: Comparing NLL across 2 large data sets.

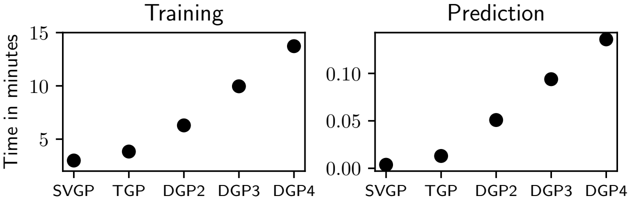

Figure 11: Average clock times for 100 runs with 1200 epochs on `energy`. Predictions use 100 samples from the posterior. The variance of training and prediction repetitions is negligible ($< 10^{-5}$).

Medium Scale Regression We compare TGPs, SVGPs and DGPs with 2, 3, 4 layers on a range of medium-sized data sets. For D_x denoting the dimension of $\mathbf{X}$, each DGP layer has at least $\min(D_x, 16)$ GP's per layer, and we set kernel parameters and perform inference as in Salimbeni and Deisenroth (2017). Results over 10 training:test (90:10) splits are shown in

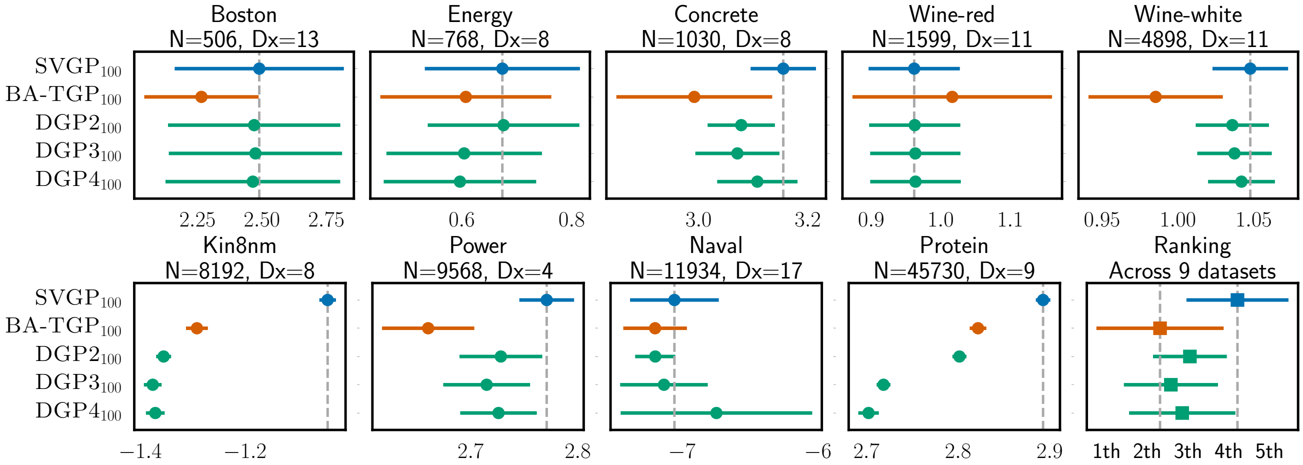

Figure 12: Comparing NLL (left is better) across 9 data sets. **Bottom rightmost panel:** Ranking of the methods across all 9 data sets and repetitions shows that the TGP performs as well as a 3- or 4-layer DGP.

Fig. 12. They demonstrate that the TGP clearly outperforms the GP and, on 5/9 datasets, the TGP even manages to outperform the 4-layer DGP. When ranking the results, this implies an overall performance comparable to that of a 3-layer DGP (see the rightmost bottom plot in Fig. 12).

Large Scale Regression We also benchmark the TGP on two large scale regression datasets. The `Year` has 0.5M, and the `airline` around 2M training data points. Fig. 10 shows the results and mirrors the findings for the medium-sized regressions.

Classification Unlike transformations of the likelihood, prior transformations can be used to improve performance for discrete-valued data. Fig. 13 makes this point using a range of classification problems.

Computational Cost Impressively, the TGP can match the 3-layer DGP's performance at a fraction of the computational complexity: Even on the `energy` data set—where the DGP has only 8 GPs per layer—computation times for the 3-layer DGP are 3× (training) and 10× (prediction) that of the TGP (Fig. 11). This is true even though our implementation of the DGP fully exploits parallelization on a GPU cluster, while our current TGP implementation does not exploit potential parallelization across GP parameters. We further emphasize that our predictions use a 100 point quadrature integration rule per Monte Carlo sample, i.e we use 100 × 100 integration points, see App. A.4, in contrast to the DGP—which only uses 100.

Fewer Inducing Points The TGP can match or outperform standard GPs using 20 times fewer inducing points, and we illustrate this in Fig. 9. We provide evidence on additional data sets in App. B which also shows that $\mathrm{Cov}[q(\mathbf{f}_0)]$ does not collapse to a point mass. This implies that the Bayesian NN flow does not make the GP redundant—in fact, the TGP's base GP is essential to our model's quantification of uncertainty.

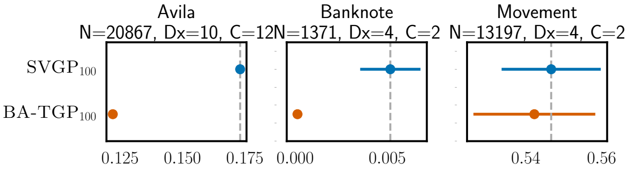

Figure 13: Comparing NLL (left is better) on three classification problems with up to $C = 12$ classes.

6 Conclusions and Future Work

We introduced Transformed Gaussian Processes (TGPs)—a new and flexible family of function priors—by enriching GPs with parameterized, invertible, input-dependent, and Bayesian transformations (aka flows). While the computational overhead of our inference scheme is comparable to that of sparse variational GP regression, its predictive performance matches that of multi-layered DGPs. The variational approximation we derived also speeds up inference in a host of other models (e.g. Wilson and Ghahramani, 2010; Wauthier and Jordan, 2010; Snelson et al., 2003) to $\mathcal{O}(NM^2 + M^3)$. Our work can be used within inter-domain inducing point approximations (Lázaro-Gredilla and Figueiras-Vidal, 2009; Dutordoir et al., 2020), to improve multitask GPs (Bonilla et al., 2007; Álvarez et al., 2012; Hamelijnck et al., 2019), density estimation (Dutordoir et al., 2018), model calibration (Maroñas et al., 2020), and probabilistic dimensionality reduction (Titsias and Lawrence, 2010).

Acknowledgments

J. M. is supported by grant FPI-UPV under grant agreement No 825111 DeepHealth Project, and by the Spanish National Ministry of Education through grant RTI2018-098091-B-I00. O. H., J. K., and T. D. are funded by the Lloyd's Register Foundation programme on Data Centric Engineering through the London Air Quality project. O. H. is funded through The Alan Turing Institute PhD fellowship programme. J. K. is funded by the EPSRC grant EP/L016710/1 as part of the Oxford-Warwick Statistics Programme (OxWaSP) and the Facebook Fellowship Programme. TD acknowledges support from UKRI Turing AI Fellowship EP/V02678X/1. This work is also supported by EPSRC grant EP/T004134/1, by The Alan Turing Institute for Data Science and AI under EPSRC grant EP/N510129/1 in collaboration with the Greater London Authority, by the PRHLT Research Center and by the European Union under project Sistemas de frabricación inteligentes para la industria 4.0 – grant agreement IDIFEDER/2018/025.

References

Bauer, M., van der Wilk, M., and Rasmussen, C. E. (2016). Understanding probabilistic sparse gaussian process approximations. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 1533–1541. Curran Associates, Inc.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Bonilla, E. V., Chai, K. M. A., and Williams, C. K. I. (2007). Multi-task gaussian process prediction. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, page 153–160, Red Hook, NY, USA. Curran Associates Inc.

Box, G. E. P. and Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 26(2):211–252.

Calandra, R., Peters, J., Rasmussen, C. E., and Deisenroth, M. P. (2016). Manifold gaussian processes for regression. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 3338–3345.

Chen, S. S. and Gopinath, R. A. (2000). Gaussianization. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS'00, page 402–408, Cambridge, MA, USA. MIT Press.

Damianou, A. and Lawrence, N. (2013). Deep gaussian processes. In Carvalho, C. M. and Ravikumar, P., editors, *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of Machine Learning Research*, pages 207–215, Scottsdale, Arizona, USA. PMLR.

Dezfouli, A. and Bonilla, E. V. (2015). Scalable inference for gaussian process models with black-box likelihoods. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1414–1422, Cambridge, MA, USA. MIT Press.

Diggle, P. and Ribeiro, P. (2007). *Model-based Geostatistics*. Springer Series in Statistics. Springer.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using real nvp.

Dubois, G. (2003). *Mapping radioactivity in the environment : Spatial Interpolation Comparison 97*. Office for Official Publications of the European Communities, Luxembourg.

Dutordoir, V., Durrande, N., and Hensman, J. (2020). Sparse gaussian processes with spherical harmonic features. In *International Conference on International Conference on Machine Learning*.

Dutordoir, V., Salimbeni, H., Hensman, J., and Deisenroth, M. (2018). Gaussian process conditional density estimation. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 2385–2395. Curran Associates, Inc.

Einstein, A. (1905). Über die von der molekularkinetischen theorie der wärme geforderte bewegung von in ruhenden flüssigkeiten suspendierten teilchen. *Annalen der physik*, 4.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1050–1059. JMLR.org.

Gardner, J. R., Pleiss, G., Bindel, D., Weinberger, K. Q., and Wilson, A. G. (2018). Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*.

Garnelo, M., Schwarz, J., Rosenbaum, D., Viola, F., Rezende, D. J., Eslami, S. M. A., and Teh, Y. W.

(2018). Neural processes. In *Workshop on Theoretical Foundations and Applications of Deep Generative Models, International Conference on Machine Learning*.

Hamelijnck, O., Damoulas, T., Wang, K., and Girolami, M. (2019). Multi-resolution multi-task gaussian processes. In *Advances in Neural Information Processing Systems*, pages 14025–14035.

Hartmann, M. and Vanhatalo, J. (2019). Laplace approximation and natural gradient for gaussian process regression with heteroscedastic student-t model. *Statistics and Computing*, 29(4):753–773.

Hensman, J., de G. Matthews, A. G., and Ghahramani, Z. (2015). Scalable variational gaussian process classification. In Lebanon, G. and Vishwanathan, S. V. N., editors, *AISTATS*, volume 38 of *JMLR Workshop and Conference Proceedings*. JMLR.org.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, page 282–290, Arlington, Virginia, USA. AUAI Press.

Jones, M. C. and Pewsey, A. (2009). Sinh-arcsinh distributions. *Biometrika*, 96(4):761–780.

Jylänki, P., Vanhatalo, J., and Vehtari, A. (2011). Robust gaussian process regression with a student-t likelihood. *Journal of Machine Learning Research*, 12(11).

Knoblauch, J. (2019). Robust deep gaussian processes. *arXiv preprint arXiv:1904.02303*.

Knoblauch, J., Jewson, J., and Damoulas, T. (2019). Generalized variational inference: Three arguments for deriving new posteriors. *arXiv preprint arXiv:1904.02063*.

Krige, D. G. (1951). *A statistical approach to some mine valuation and allied problems on the Witwatersrand: By DG Krige*. PhD thesis, University of the Witwatersrand.

Lázaro-Gredilla, M. (2012). Bayesian warped gaussian processes. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1619–1627. Curran Associates, Inc.

Lázaro-Gredilla, M. and Figueiras-Vidal, A. (2009). Inter-domain gaussian processes for sparse inference using inducing features. In Bengio, Y., Schuurmans, D., Lafferty, J. D., Williams, C. K. I., and Culotta, A., editors, *Advances in Neural Information Processing Systems 22*, pages 1087–1095. Curran Associates, Inc.

Lichman, M. (2013). UCI machine learning repository.

Lin, L.-H. and Joseph, V. R. (2019). Transformation and additivity in gaussian processes. *Technometrics*, 0(0):1–11.

London, I. C. (2020). Londonair - london air quality network (laqn). `https://www.londonair.org.uk`.

Maroñas, J., Paredes, R., and Ramos, D. (2020). Calibration of deep probabilistic models with decoupled bayesian neural networks. *Neurocomputing*, 407:194–205.

Meng, C., Song, Y., Song, J., and Ermon, S. (2020). Gaussianization flows. volume 108 of *Proceedings of Machine Learning Research*, pages 4336–4345, Online. PMLR.

Monterrubio-Gómez, K., Roininen, L., Wade, S., Damoulas, T., and Girolami, M. (2020). Posterior inference for sparse hierarchical non-stationary models. *Computational Statistics & Data Analysis*, page 106954.

Murray, I., MacKay, D., and Adams, R. P. (2009). The gaussian process density sampler. In *Advances in Neural Information Processing Systems*, pages 9–16.

Muré, J. (2018). Trans-gaussian kriging in a bayesian framework : a case study. *arXiv: Applications*.

O'Hagan, A. (1978). Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society: Series B (Methodological)*, 40(1):1–24.

Oliveira, V. D., Kedem, B., and Short, D. A. (1997). Bayesian prediction of transformed gaussian random fields. *Journal of the American Statistical Association*, 92(440):1422–1433.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Quiñonero Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate gaussian process regression. *J. Mach. Learn. Res.*, 6:1939–1959.

Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France. PMLR.

Rios, G. (2020). Transport gaussian processes for regression.

Rios, G. and Tobar, F. (2019). Compositionally-warped gaussian processes. *Neural Networks*, 118:235–246.

Salimbeni, H. and Deisenroth, M. (2017). Doubly stochastic variational inference for deep gaussian processes. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 4588–4599. Curran Associates, Inc.

Sklar, A. (1959). Fonctions de répartition à n dimensions et leurs marges. *Publications de l'Institut de Statistique de l'Université de Paris*, 8:229–231.

Snelson, E., Rasmussen, C. E., and Ghahramani, Z. (2003). Warped gaussian processes. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, page 337–344, Cambridge, MA, USA. MIT Press.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Titsias, M. (2009). Variational learning of inducing variables in sparse gaussian processes. volume 5 of *Proceedings of Machine Learning Research*, pages 567–574, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA. PMLR.

Titsias, M. and Lawrence, N. D. (2010). Bayesian gaussian process latent variable model. volume 9 of *Proceedings of Machine Learning Research*, pages 844–851, Chia Laguna Resort, Sardinia, Italy. JMLR Workshop and Conference Proceedings.

Tsai, A. C., Liou, M., Simak, M., and Cheng, P. E. (2017). On hyperbolic transformations to normality. *Computational Statistics & Data Analysis*, 115:250 – 266.

Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley.

Turner, R. E. and Sahani, M. (2011). *Two problems with variational expectation maximisation for time series models*, page 104–124. Cambridge University Press.

Uhlenbeck, G. E. and Ornstein, L. S. (1930). On the theory of the brownian motion. *Physical review*, 36(5):823.

Ustyuzhaninov, I., Kazlauskaite, I., Kaiser, M., Bodin, E., Campbell, N., and Henrik Ek, C. (2020). Compositional uncertainty in deep gaussian processes. volume 124 of *Proceedings of Machine Learning Research*, pages 480–489, Virtual. PMLR.

Wauthier, F. L. and Jordan, M. I. (2010). Heavy-tailed process priors for selective shrinkage. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 2406–2414. Curran Associates, Inc.

Williams, C. K. and Rasmussen, C. E. (1996). Gaussian processes for regression. In *Advances in neural information processing systems*, pages 514–520.

Williams, C. K. I. and Seeger, M. (2001). Using the nyström method to speed up kernel machines. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press.

Wilson, A. G. and Ghahramani, Z. (2010). Copula processes. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 2460–2468. Curran Associates, Inc.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. (2016). Deep kernel learning. volume 51 of *Proceedings of Machine Learning Research*, pages 370–378, Cadiz, Spain. PMLR.

Álvarez, M. A., Rosasco, L., and Lawrence, N. D. (2012). Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266.