Residual Deep Gaussian Processes on Hyperspheres

Kacper Jakub Wyrwal

4th Year Project Report
Artificial Intelligence and Computer Science
School of Informatics
University of Edinburgh

2024

Abstract

Gaussian processes are a powerful probabilistic tool for learning unknown functions, particularly in tasks such as Bayesian optimization, active learning, and reinforcement learning where accurate uncertainty estimates are crucial. However, data in these tasks often has inherent geometric structure, residing on non-Euclidean manifolds. While various Gaussian process constructions have been developed to harness the geometry of data for improved predictions, they are often limited by the simplicity bias of their defining kernels. Residual deep Gaussian processes aim to alleviate this limitation by sequentially combining multiple Gaussian processes, but little has been known about their performance on Riemannian manifolds.

This thesis investigates residual deep Gaussian processes, focusing on their strengths, weaknesses, and applicability to various tasks on hypersphere domains. Through a series of experiments, the impact of model depth and data density on performance is evaluated using synthetic functions designed to test the model's ability to capture irregular patterns. The effectiveness of residual deep Gaussian processes in Bayesian optimization of irregular functions is demonstrated, highlighting a symbiotic approach that combines shallow Gaussian processes for initial exploration with deep models for subsequent exploitation. Furthermore, the applicability of the model to Euclidean data is explored by projecting real-world datasets onto hyperspheres and employing a specialized variational inference strategy based on spherical harmonics.

The results show that residual deep Gaussian processes can significantly outperform shallow Gaussian processes in modeling irregular functions when sufficient training data is available, with performance increasing with depth up to a saturation point. The model exhibits superior median performance and remarkable stability compared to a simplified deep Gaussian process baseline. However, in sparse data regimes, the performance of residual deep Gaussian processes deteriorates with increasing depth. The findings also suggest that manifold learning techniques can enhance the applicability and performance of residual deep Gaussian processes when dealing with Euclidean data.

This thesis contributes to the growing body of knowledge on residual deep Gaussian processes by demonstrating their effectiveness in regression and Bayesian optimization tasks, identifying their strengths and weaknesses in various data regimes, and exploring their application to Euclidean data. Looking ahead, our findings open up promising avenues for future research, such as more elaborate adaptive strategies for Bayesian optimization that periodically switch between shallow and deep models to further enhance performance on irregular functions with multiple local optima. Moreover, the recently introduced intrinsic Gaussian vector field construction presents an exciting opportunity to apply residual deep Gaussian processes to complex real-world problems like modeling wind velocities near the Earth's surface.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Kacper Jakub Wyrwal*)

Acknowledgements

I would like to express my most sincere gratitude to my supervisors Viacheslav Borovitskiy and Edoardo Ponti, without whom none of this would be possible.

Table of Contents

Chapter 1

Introduction

Gaussian processes [Rasmussen and Williams, 2006] are a powerful, principled approach for modelling unknown functions in the Bayesian framework. They are known for their ability to incorporate prior information about the target function, resilience to overfitting, and well-calibrated uncertainty estimates. This makes them common models of choice for reinforcement learning [Snoek et al., 2012], Bayesian optimisation [Deisenroth and Rasmussen, 2011, Jaquier et al., 2021], active learning [Srinivas et al., 2012], etc. Gaussian processes also garner interest in deep learning research due to their equivalence to classes of infinitely wide or infinitely deep neural networks [Jacot et al., 2018, Eleftheriadis et al., 2023]. While Gaussian processes have proven to be effective in these domains, they typically implicitly assume that the data they operate on lies in a Euclidean space. However, in many real-world problems, data exhibit complex geometric structures that deviate from the Euclidean setting, necessitating the development of Gaussian processes that can adapt to these non-Euclidean domains.

Recent years in machine learning research have seen increased interest in utilising the geometric structure of data to improve the design and performance of machine learning models. In many problems, the data lies on spaces with geometric properties different from those of the Euclidean space. There is an abundance of domains to which data naturally adhere; from simple domains such as undirected, unweighted graphs to domains rich with additional information like curvature in differentiable, or Riemannian, manifolds. Indeed, sometimes the geometric properties of the data are unknown a priori. However, even then, methods based on *manifold learning*, which have seen promising new developments [Fichera et al., 2023], can help to uncover the geometric structure of data.

Gaussian processes too have seen a flourishing of research directed at harnessing the geometry of data. Generalisations of Gaussian processes and methods for their efficient implementation have been proposed for a variety of non-Euclidean domains, including Riemannian manifolds [Borovitskiy et al., 2020], Lie groups [Azangulov et al., 2022, 2023], graphs [Coveney et al., 2020] and cellular complexes [Alain et al., 2023]. This has also been combined with ideas from manifold learning to produce Gaussian processes that implicitly learn the geometric structure of data [Fichera et al., 2023]. By exploiting the explicit or implicit geometric properties of data, these methods have been shown to yield improvements across many areas of application and different tasks, including regression in climate science Hutchinson et al. [2021], Robert-Nicoud et al. [2023], physical systems modelling [Hutchinson et al., 2021], and other standard benchmarks [Jaquier and Rozo, 2020], as well as Bayesian optimisation and reinforcement learning in a variety of robotics tasks [Jaquier and Rozo, 2020, Jaquier et al., 2021].

Despite the advancements in Gaussian processes for non-Euclidean spaces, standard Gaussian processes may still struggle to model complex, irregular functions due to their bias towards simple functions arising from common kernel choices. One approach to address this issue is through more expressive kernels, which has been attempted from compositional [Duvenaud et al., 2013] and parameterised [Wilson et al., 2015] perspectives. Another solution, inspired by the success of deep learning, is to sequentially combine multiple Gaussian processes, forming a *deep Gaussian process* [Damianou and Lawrence, 2013]. The deep, layered structure of these models has been shown to improve performance in a variety of complex regression and classification tasks involving large datasets [Salimbeni and Deisenroth, 2017].

However, until recently, an appropriate generalisation of deep Gaussian processes to non-Euclidean spaces had been missing. To address this problem, during a semester research project at ETH Zurich, we utilised the recent advances in vector-valued Gaussian processes [Robert-Nicoud et al., 2023, Hutchinson et al., 2021] to propose a new class of models named *residual deep Gaussian processes*. These models generalise the class of Euclidean deep Gaussian processes from the work of Salimbeni and Deisenroth [2017] and work with data on Riemannian manifolds. They are particularly appealing because they respect the geometry of data without sacrificing many useful properties that their Euclidean counterparts enjoy, including efficient approximate training and inference, as well as efficient approximate sampling of functions from their posterior [Wilson et al., 2020a,b].

Although these properties painted a promising picture of residual deep Gaussian processes, our preliminary experimental examination of these models was very limited and did not definitely establish that these models offer an advantage over standard Gaussian processes. In fact, in most cases, our model was outperformed by standard Gaussian processes, and only in one experiment, where data were abundant enough, did we see residual deep Gaussian processes offer a modest advantage.

In this fourth year project, we undertook to obtain a comprehensive picture of residual deep Gaussian processes through a range of experiments in regression and Bayesian optimisation, on stylised examples, standard benchmarks, and real-world data. We specifically chose to focus our attention on hyperspheres as the domain class for our model. We focus on one class of manifolds to make our investigation more focused, while we chose hyperspheres for their convenient properties, such as compactness, and their application to regression on Euclidean data thanks to Dutordoir et al. [2020]. Through this examination, we obtained a multi-faceted understanding of our model, showing its strengths, weaknesses, as well as future directions for research and improvements.

1.1 Contributions

In this work, we conducted a range of experiments in order to develop a multi-faceted understanding of the strengths and weaknesses of residual deep Gaussian processes, specifically in the setting of hyperspheres. Indeed, we have broadened our understanding of these models, showing that they can outperform shallow Gaussian processes in tasks on hyperspheres and on data projected onto a hypersphere when data is abundant enough, as well as demonstrating success in Bayesian optimisation of irregular functions on hyperspheres. Concretely, our contributions are as follows.

- We evaluated residual deep Gaussian processes on a synthetic regression task on the sphere with double the number of data points over our previous experiments owing to a new, vectorised implementation. With these experiments, we demonstrated that our model can significantly outperform standard Gaussian processes in modelling irregular functions on the sphere. Moreover, in contrast to our previous work, we showed that, with sufficient data, model performance can increase monotonically from 1 to 4 hidden layers.

- We introduced a simple baseline deep Gaussian process on Riemannian manifolds, built from an initial geometry-aware Gaussian process composed with a Euclidean deep Gaussian process. We then applied this baseline model to the regression task on an irregular function described above and compared its performance to residual deep Gaussian processes. We demonstrated that residual deep Gaussian processes achieve superior median performance across all model depths in terms of test log-likelihood. However, we also showed that the performance of the baseline varies significantly more than that of our model and thus in individual experiment runs can noticeably outperform residual deep Gaussian processes.

- We evaluated residual deep Gaussian processes for Bayesian optimisation of functions on hyperspheres. We showed that for irregular objective functions, our model can offer significant improvements in the quality of optima found. Moreover, we demonstrated that when the set of initial observations is sparse, our model can be used in tandem with shallow Gaussian processes to avoid problems due to data sparsity. We also confirmed that for Bayesian optimisation of a function smooth around the optimum, residual deep Gaussian processes offer no noticeable advantage over standard Gaussian processes.

- We evaluated our model on six real-world regression datasets with Euclidean data from the UCI Machine Learning Repository [Kelly et al., 2023]. To this end, we followed Dutordoir et al. [2020], projecting the data onto a hypersphere and applying a specific class of approximate Gaussian processes based on the spherical harmonics. Training on the sphere-projected data, we demonstrated that residual deep Gaussian processes can offer an advantage over standard Gaussian processes, and this advantage seems to increase with the size of the dataset. Finally, we performed the same experiments with Euclidean deep Gaussian processes directly on the datasets and found that only on one of the datasets did our results match those reported by Salimbeni and Deisenroth [2017], while on others the performance exhibits anomalous patterns and often a collapse, inviting additional research to understand this phenomenon.

1.2 Report Structure

Our project report begins with a motivation for and introduction to a confluence of topics in Gaussian processes that come together to make residual deep Gaussian processes. We begin with an exposition of the topics in Euclidean space, detailing classical Gaussian processes, approximate Gaussian processes, efficient posterior sampling, and deep Gaussian processes. Then, we detail their generalisation to Riemannian manifolds, beginning with scalar-valued Gaussian processes, through vector-valued Gaussian processes, and culminating at deep Gaussian processes.

The remaining part of the report follows three experiments in regression and Bayesian optimisation ordered approximately by the complexity of the tasks involved. We start with a simple task modelling an irregular function on the sphere in Chapter 3, then, in Chapter 4, we evaluate our model in finding the optima of irregular functions through Bayesian optimisation, and, in Chapter 5, we attempt inference on Euclidean data projected on hyperspheres. We summarise our findings from these experiments in Chapter 6 and indicate promising avenues for future research.

Below we give the structure of our report in more detail.

1. In Chapter 2, we give a background on Gaussian processes in order to make the construction of residual deep Gaussian processes clear and intuitive. We introduce and motivate Gaussian processes, describe how to perform approximate inference Rasmussen and Williams [2006], and detail the construction of deep Gaussian processes and the respective doubly-stochastic variational inference technique [Salimbeni and Deisenroth, 2017]. We also detail how to build Gaussian processes on Riemannian manifolds using the generalised Matern kernel Borovitskiy et al. [2020].

2. In Chapter 3, we detail the construction of residual deep Gaussian processes as well as introduce a new baseline — a simple generalisation of deep Gaussian processes to manifolds based on a geometry-aware Gaussian process followed by a Euclidean deep Gaussian process. We evaluate residual deep Gaussian processes in modelling an irregular function on the sphere, showing that in the densest data regime tested our model significantly outperforms the standard Gaussian process, with performance increasing with depth. Lastly, on the same regression task, we show that residual deep Gaussian processes outperform the simpler variant of deep Gaussian processes on manifolds,

3. In Chapter 4, we give a brief introduction to Bayesian optimisation, motivating its geometry-aware variant, as well as detailing technical aspects of performing Bayesian optimisation with deep Gaussian processes. We then apply residual deep Gaussian processes to two Bayesian optimisation tasks on hyperspheres, including an irregular and a smooth target function, and compare its performance to Bayesian optimisation done only with a shallow Gaussian process. We find that our model offers, often immediate, improvements over the shallow Gaussian pro-

cesses in optimising the irregular function; however, its improved expressiveness offered little advantage for the optimisation of the smooth function.

4. In Chapter 5, we detail a strategy for applying geometry-aware Gaussian processes on hyperspheres to Euclidean data. This strategy hinges on projecting the data onto a hypersphere and applying a variational approximation to the Gaussian process based on inner products with spherical harmonics proposed by Dutordoir et al. [2020]. We evaluated residual deep Gaussian processes in this setting on a selection of datasets from the UCI Machine Learning Repository, showing improvement over standard Gaussian processes utilising the scheme on many of the tested datasets, for large enough datasets. We attempted to compare our model to Euclidean deep Gaussian processes applied directly to the data; however, due to their instability between experiment runs and performance collapse at higher levels the comparison was unclear.

5. In Chapter 6, we summarize the findings, bringing them together to form a unifying picture of our investigation. We also indicate directions for future work, including an investigation of new strategies for Bayesian optimisation by adaptively choosing between shallow Gaussian processes and deep residual Gaussian processes. We suggest that a recently introduced construction of intrinsic Gaussian vector fields could be applied with residual deep Gaussian processes to model wind velocities close to the Earth's surface — a task with which shallow Gaussian vector fields may struggle in certain cases.

Chapter 2

Background

In this chapter, we present the necessary background needed to understand the construction of deep Gaussian processes on Riemannian manifolds.

We first build our way up from standard Gaussian processes to deep Gaussian processes in Euclidean space. We begin with a brief introduction and motivation for Gaussian processes in general. We then review how to perform inference with Gaussian processes efficiently when the number of data points is large, and how to efficiently sample functions from Gaussian process posteriors. Finally, we give an exposition of deep Gaussian processes — a multi-layer generalisation of Gaussian processes. These deep models are more expressive than standard Gaussian processes [Salimbeni and Deisenroth, 2017], akin to how neural networks are more expressive than linear models [Goodfellow et al., 2016], which can be an advantage on complex tasks.

Afterwards, we turn our attention to Gaussian processes on Riemannian manifolds, detailing how to construct these models in a computationally viable way using a generalisation of the Matérn class of kernels.

2.1 Gaussian Processes

In supervised learning settings, which are the focus of our work, we hope to estimate the true data-generating process $y : \mathbb{R}^d \to \mathbb{R}$ utilising a labelled training dataset $\mathcal{D} = (x_i, y_i)_{i=1}^n$. In other words, from the finite training set, we try to construct a function f that approximates the true function y.

To make progress on this task, we need to make assumptions about f; otherwise, all functions compatible with the training data would be equally good candidates. Often, such assumptions are made by restricting the search space to a specific class of functions, for example linear functions or functions defined by neural networks. An alternative approach is to assign a prior probability to every possible function using stochastic processes, which are, intuitively speaking, distributions over functions.

A Gaussian process is a stochastic process that generalizes the Gaussian distribution to distributions over functions. It is particularly attractive for its simple form and

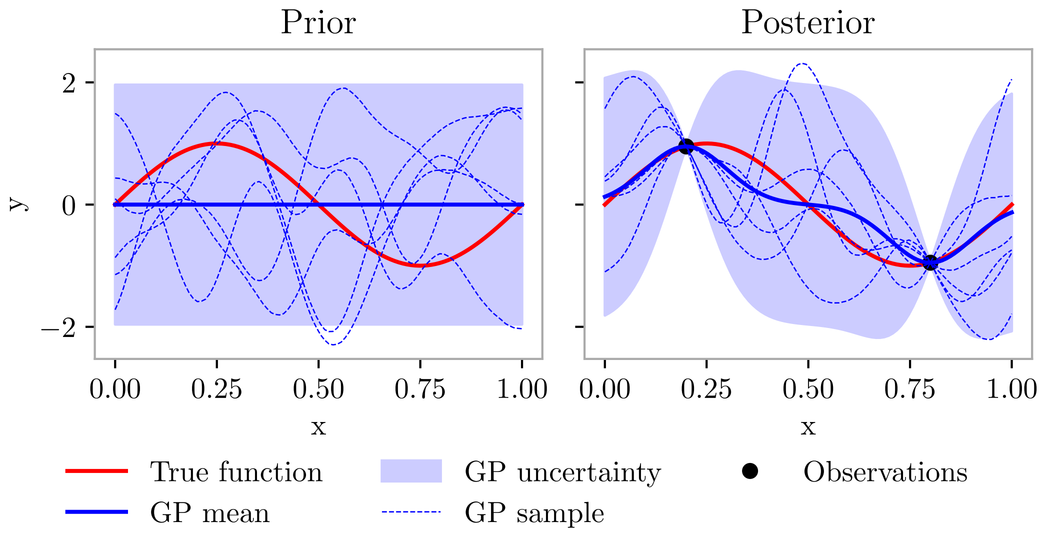

Figure 2.1: Prior (Left) and posterior (right) distribution over functions according to a Gaussian process.

mathematical elegance in many of its properties. Formally, a random function $f : \mathbb{R}^d \to \mathbb{R}$ is distributed according to a Gaussian process with a mean function $\mu : \mathbb{R}^d \to \mathbb{R}$ and covariance function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ if, for any finite set of points $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{R}^{d\times n}$, we have

$$f(\mathbf{X}) \sim \mathcal{N}(\mu(\mathbf{X}), k(\mathbf{X}, \mathbf{X})), \tag{2.1}$$

where $f(\mathbf{X})_i = f(\mathbf{x}_i)$, $\mu(\mathbf{X})_i = \mu(\mathbf{x}_i)$, and $k(\mathbf{X}, \mathbf{X})_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$. That is, every finite marginal of f follows a multivariate Gaussian distribution with mean and covariance defined by μ and k as described above. If this is the case, we write

$$f \sim \mathcal{GP}(\mu, k). \tag{2.2}$$

To use Gaussian processes for inference in a supervised learning setting, we start with a prior Gaussian process $f \sim \mathcal{GP}(\mu, k)$, which incorporates our initial beliefs about the target function y. We then update our beliefs based on evidence - that is, *condition* f on observations $\mathbf{y} = (y_1, \dots, y_n) \in \mathbb{R}^n$ at a set of points $\mathbf{X} = (x_1, \dots, x_n) \in \mathbb{R}^{d\times n}$. The appeal of Gaussian processes comes in part from the fact that conditioned, or *posterior*, Gaussian processes are Gaussian processes themselves and have a simple closed form

$$\begin{aligned} f(\cdot)|\mathbf{y};\mathbf{X} = \mathcal{GP}(&k(\cdot,\mathbf{X})k(\mathbf{X},\mathbf{X})^{-1}(\mathbf{y} - \mu(\mathbf{X})), \\ &k(\cdot,\cdot) - k(\cdot,\mathbf{X})k(\mathbf{X},\mathbf{X})^{-1}k(\mathbf{X},\cdot)). \end{aligned} \tag{2.3}$$

Typically, the observations $\mathbf{y}$ are noisy corruptions of the true unobserved data-generating process. In this case, in addition to f, which is an approximation to the unobserved process, we use a likelihood which factorises over observations

$$p(\mathbf{y}|f(\mathbf{X})) = \prod_{i=1}^{n} p(y_i|f(\mathbf{x}_i)). \tag{2.4}$$

If the likelihood is Gaussian

$$p(y_i|f(\mathbf{x}_i)) = \mathcal{N}(y_i|f(\mathbf{x}_i),\sigma^2), \qquad \sigma \in (0,\infty) \tag{2.5}$$

then the posterior Gaussian process conditioned on the noisy observations is also a Gaussian process with a simple form

$$\begin{aligned} f(\cdot)|\mathbf{y};\mathbf{X} = \mathcal{GP}(&k(\cdot,\mathbf{X})(k(\mathbf{X},\mathbf{X})+\sigma^2\mathbf{I})^{-1}(\mathbf{y}-\mu(\mathbf{X})), \\ &k(\cdot,\cdot)-k(\cdot,\mathbf{X})(k(\mathbf{X},\mathbf{X})+\sigma^2\mathbf{I})^{-1}k(\mathbf{X},\cdot)). \end{aligned} \tag{2.6}$$

Vector-valued Gaussian processes, used for modeling vector-valued functions, are defined in the same way, except that the mean function is vector valued and the covariance function is matrix-valued - giving the covariance matrix between values of the Gaussian process (which are themselves random vectors) at two points. In the Euclidean case, these are typically implemented by stacking multiple scalar-valued Gaussian processes in a vector (in this case the values of the covariance function are diagonal matrices, as the Gaussian processes in each dimension are independent from each other), but more sophisticated methods exist [Bonilla et al., 2007].

When employing Gaussian processes, we choose the mean μ and covariance k functions to incorporate prior knowledge about the function we are trying to model. While the choice of the mean function is essentially unrestricted, the covariance function must be *positive semi-definite*, meaning that for all finite subsets of points $\mathbf{X} \in \mathbb{R}^{d\times n}$ the corresponding covariance matrix $k(\mathbf{X},\mathbf{X})$ must be positive semi-definite, i.e.,

$$\mathbf{z}^T\mathbf{k}(\mathbf{X},\mathbf{X})\mathbf{z} \geq 0 \quad \forall \mathbf{z} \in \mathbb{R}^n. \tag{2.7}$$

The Matérn class of covariance functions has been widely adopted in practice owing to its computational simplicity and useful properties such as stationarity

$$k(\mathbf{x},\mathbf{y}) = k(\mathbf{x}+\mathbf{z},\mathbf{y}+\mathbf{z}) \quad \forall;\mathbf{x},\mathbf{y},\mathbf{z} \in \mathbb{R}^d \tag{2.8}$$

and a controllable degree of smoothness. It has a few convenient parameters specifying properties of the corresponding Gaussian process $f \sim \mathcal{GP}(\mu,k)$

- $\sigma^2 \in (0,\infty)$, or *output-scale*, determines the point-wise variance of f, i.e., $\mathrm{Var}[f(\mathbf{x})]$.
- $\kappa \in (0,\infty)$, or *length-scale*, regulates how quickly $\mathrm{Cov}[f(\mathbf{x}),f(\mathbf{x}')]$ decays as $||\mathbf{x}-\mathbf{x}'||$ increases.
- $\nu \in (0,\infty]$, or *smoothness*, specifies the smoothness of f, where higher ν indicates a higher degree of differentiability.

These parameters can be inferred from data, for instance via gradient descent maximising the likelihood of the observations. In Euclidean spaces, the Matérn covariance function is given by the formula

$$k_{\nu,\kappa,\sigma^2}(\mathbf{x},\mathbf{x}') = \begin{cases} \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{||\mathbf{x}-\mathbf{x}'||}{\kappa}\right)^\nu K_\nu\left(\sqrt{2\nu}\frac{||\mathbf{x}-\mathbf{x}'||}{\kappa}\right) & \text{if } \nu \in (0,\infty) \\ \sigma^2 \exp\left(-\frac{||\mathbf{x}-\mathbf{x}'||^2}{2\kappa^2}\right), & \text{if } \nu = \infty, \end{cases} \tag{2.9}$$

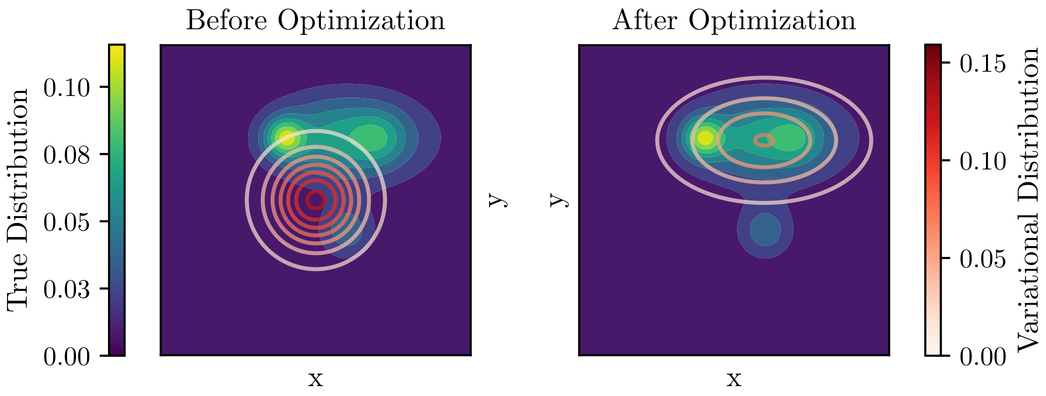

Figure 2.2: Schematic illustration of variational inference. A complex true 2d distribution (shown using filled-in contours) is approximated by a simple 2d variational distribution from the family of multivariate Gaussian distributions with diagonal covariance matrices. *Left:* The true distribution and an initial state of a variational distribution prior to optimisation. *Right:* The true and variational distributions after optimisation by minimising the Kullback-Leibler divergence between the two distributions.

where Γ is the gamma function, and K_ν is the modified Bessel function of the second kind.

The Matérn kernel with $\nu = \infty$ is commonly known as the squared exponential or radial basis function (RBF) kernel. It is infinitely differentiable, which implies that functions supported by the RBF-kernel Gaussian process have derivatives of all orders.

The Matérn class of kernels is especially important for our work, as its generalisation to manifolds underpins much of the recent development in geometry-aware Gaussian processes and, consequently, our construction of residual deep Gaussian processes. In our experiments, we used both the Matérn kernels with $\nu < \infty$ (Chapter 3 and Chapter 4) and the RBF kernel (Chapter 5) to match the existing literature.

2.2 Approximate Inference

In practice, inference with Gaussian processes using Equation (2.6) can be computationally prohibitive. The asymptotic bottleneck comes from the matrix inversion $k(\mathbf{X},\mathbf{X})^{-1}$, which has cubic complexity in the number of observations conditioned on. Numerous ways of addressing this issue have been introduced, including low-rank approximations [Quiñonero-Candela and Rasmussen, 2005], greedy methods [Seeger et al., 2003], and other specialized techniques [Hensman et al., 2013].

In this report, we focus on a widely adopted approximation technique based on inducing points Rasmussen and Williams [2006], van der Wilk et al. [2020]. This is a simple, effective, and efficient inference method based on a variational approximation to the exact Gaussian process posterior.

Specifically, in the inducing points method, we define a set of locations $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_m) \in \mathbb{R}^{d\times m}$ and a variational distribution $q(\mathbf{U}) = \mathcal{N}(\mathbf{U}|\mathbf{m},\mathbf{S})$ that defines the distribution of our Gaussian process at $\mathbf{Z}$. Our goal is to optimise $\mathbf{m}$, $\mathbf{S}$, and $\mathbf{Z}$ so that the variational posterior

$$q(\cdot, \mathbf{U}|\mathbf{y}; \mathbf{X}, \mathbf{Z}) = p(\cdot|\mathbf{U}; \mathbf{Z})q(\mathbf{U}; \mathbf{Z}) \tag{2.10}$$

resembles, as closely as possible, the true posterior

$$p(\cdot, \mathbf{U}|\mathbf{y}; \mathbf{X}, \mathbf{Z}) = p(\cdot|\mathbf{U}, \mathbf{y}; \mathbf{X}, \mathbf{Z})p(\mathbf{U}|\mathbf{y}; \mathbf{X}, \mathbf{Z}). \tag{2.11}$$

Matching the terms in equations Equation (2.11) and Equation (2.10), we are trying to achieve two things:

1. Approximate the true posterior $p(\mathbf{U}|\mathbf{y}; \mathbf{X}, \mathbf{Z})$ with our variational posterior $q(\cdot; \mathbf{X}, \mathbf{Z})$.
2. Set the locations $\mathbf{Z}$ so that $p(\cdot|\mathbf{U}, \mathbf{y}; \mathbf{X}, \mathbf{Z}) = p(\cdot|\mathbf{U}; \mathbf{Z})$. That is, we want our Gaussian process conditioned on $\mathbf{U}$ to be independent of $\mathbf{y}$. Intuitively, we want to choose to approximate the Gaussian process posterior at a set of locations $\mathbf{Z}$ that summarise all the information obtained from the observations. This can be done trivially by setting $\mathbf{Z} = \mathbf{X}$; however, typically, for improvements in inference speed, we want $\mathbf{Z}$ to be much smaller than $\mathbf{X}$.

In effect, we are transposing what was an inference problem into an optimisation problem.

Because the variational posterior is Gaussian, we can utilise the elegant mathematical properties of the Gaussian process as in Equation (2.3) to obtain a simple form for the variational posterior

$$\begin{aligned} f(\cdot)|\mathbf{U}; \mathbf{Z} \sim \mathcal{GP}(&k(\cdot, \mathbf{Z})k(\mathbf{Z}, \mathbf{Z})^{-1}(\mathbf{y} - \mathbf{m}), \\ &k(\cdot, \cdot) - k(\cdot, \mathbf{Z})k(\mathbf{Z}, \mathbf{Z})^{-1}(k(\mathbf{Z}, \mathbf{Z}) - \mathbf{S})k(\mathbf{Z}, \mathbf{Z})^{-1}k(\mathbf{Z}, \cdot)), \end{aligned} \tag{2.12}$$

though this time, in contrast to Equation (2.3), the formula for posterior covariance does not simplify further. Thus, by conditioning on a smaller set of points, the bottleneck of matrix inversion has been reduced to $O\left(m^3\right)$ and the total complexity of calculating the posterior is $O\left(m^3 + nm^2\right)$. Because m is typically much smaller than n, these variational Gaussian processes are commonly called *sparse Gaussian processes.*

The mean vector $\mathbf{m}$ and covariance matrix $\mathbf{S}$ are typically optimised by minimising the Kullback-Leibler divergence (KL divergence) between the true and the variational posterior $\mathrm{KL}(q||p)$. This is equivalent to maximizing the lower bound on the evidence (the marginal likelihood $p(\mathbf{y}|, \mathbf{U}; \mathbf{X}, \mathbf{Z})$)

$$\mathrm{ELBO} = \sum_{i=1}^{n} \mathbb{E}_{\mathbf{f}_i \sim q(\cdot|\mathbf{U})}[\log p(\mathbf{y}_i|\mathbf{f}_i)] - \mathrm{KL}(q(\mathbf{U})||p(\mathbf{U}|\mathbf{Z})). \tag{2.13}$$

As we can see in Equation (2.13), the simplicity of our choice of posterior manifests itself in the fact that the ELBO depends only on the KL-divergence between multivariate Gaussians, which has a simple closed form, and an expectation of the true log likelihood over the variational marginal posterior. Approximating the expectation with the Monte Carlo method, the complexity of evaluating the ELBO is only limited by the computation of the variational posterior.

As we shall see, this method has a natural extension to deep Gaussian processes that preserves practically all its benefits.

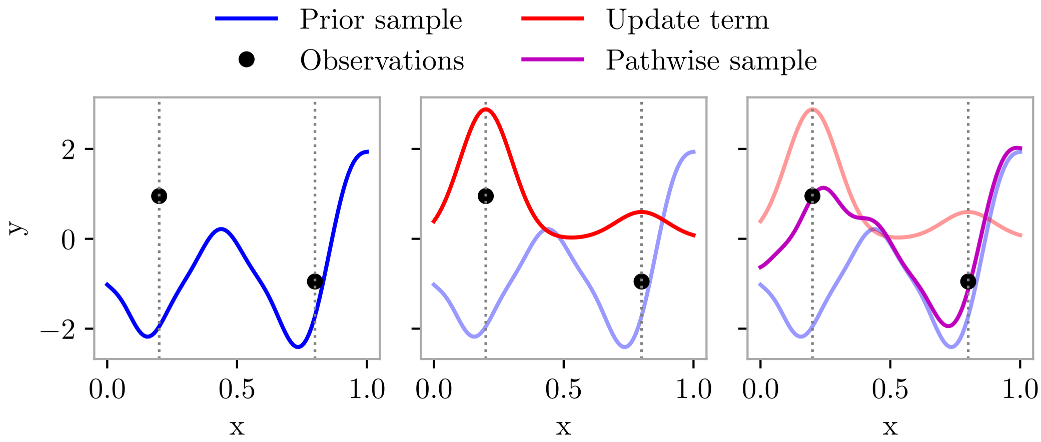

Figure 2.3: Visualization of a pathwise sample from a Gaussian process posterior. *Left:* Sample from the Gaussian process prior. *Middle:* Update the prior sample calculated using Equation (2.19). *Right:* Sum of the prior sample and update sample yields a sample from the posterior that matches the two observations.

2.2.1 Interdomain Gaussian Processes

Thus far, in using a Gaussian process $f \sim \mathcal{GP}(\mu, k)$ to model an unknown function $y : X \to \mathbb{R}$, we have only considered conditioning on observations of the values of y in form of a labelled dataset $\mathcal{D} = (\mathbf{x}_i, y_i)_{i=1}^n$. However, the value of y at a given point is not the only information we might be able to observe about y. For instance, when modelling the trajectory of an object, we may observe not only its position, but also the speed in any direction, i.e., the (directional) derivative of position.

It turns out that we can apply precisely the same formulae as for conditioning on the pointwise values of y to condition f on observations of the derivative of y at a point. In fact, because Gaussian processes are remarkably well-behaved in conjuction with linear operations, we can use the same formulae to condition f on observations of any (sufficiently well-behaved) linear transformation of y [van der Wilk et al., 2020]. Because this transformation transforms y to a different domain, we call these Gaussian processes *interdomain*.

Concretely, let $\mathrm{T} : L_2(X_1) \to L_2(X_2)$ be a linear operator and Y a set of interdomain observations $\mathcal{D} = \{\mathbf{x}_i, (\mathrm{T}y)(\mathbf{x}_i)\}_{i=1}^n$. Defining $\mathbf{X} \in \mathbb{R}^{d \times n}$ by $\mathbf{X}_i = \mathbf{x}_i$ and

$$k_{\mathrm{T}}(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{Cov}[(Tf)(\mathbf{x}_i), (Tf)(\mathbf{x}_j)], \tag{2.14}$$

the Gaussian process f conditioned on $\mathcal{D}$ is also a Gaussian process, whose distribution is given by Equation (2.6) simply by replacing k with k_{T}. Indeed, Equation (2.10) can be seen as a special case of this generalised framework, since, for the identity operator $\mathrm{T} = \mathrm{I}$, we have $k_{\mathrm{I}} = k$.

Similarly, we can generalise the variational inference method based on inducing points to interdomain Gaussian processes. This method, commonly called the inducing *variables* method, is based on a variational distribution $\mathcal{N}(\mathbf{U}|\mathbf{m}, \mathbf{S})$ which approximates the posterior distribution of $\mathrm{T}f$ at a set of interdomain inducing locations $\mathbf{Z}$, i.e., $\mathbf{S}_{i,j}$ approximates the posterior covariance $\mathrm{Cov}[(\mathrm{T}f)(\mathbf{z}_i),(\mathrm{T}f)(\mathbf{z}_j)]$ and $\mathbf{m}$ the posterior mean $\mathbb{E}[(\mathrm{T}f)(\mathbf{z}i)]$. Again, the formula for the variational posterior can be obtained from Equation (2.12) by replacing k with k_T.

This variational approximation has been used with the operator

$$\mathrm{T} : f \mapsto \langle f, \cdot \rangle \tag{2.15}$$

we can condition Gaussian processes on their inner products with other functions [Hensman et al., 2017]. In particular, Dutordoir et al. [2020] uses this operator on hyperspheres to develop a novel class of variational Gaussian processes for efficiently and uniformly accurate approximate inference on hyperspheres based on inner products with spherical harmonics. We explore this method in the context of residual deep Gaussian processes in Chapter 5.

2.2.2 Efficiently sampling functions from Gaussian process posteriors

The inducing points approximation also allows us to efficiently sample functions from sparse Gaussian process posteriors. This is a very useful property as it allows us to work with methods that rely on samples or derivatives of samples which we utilise in Chapter 4 for optimisation of acquisition functions.

Inducing points allow us to efficiently approximate Gaussian process posteriors, which is the first piece of the puzzle towards efficient sampling of functions from the posterior. The second piece is an approximation of the prior Gaussian process.

Indeed, a stationary kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ (see Equation (2.8)) can be approximated by a finite sum of the form

$$k(\mathbf{x},\mathbf{x}') = \sum_{i=1}^{n} \phi_i(\mathbf{x})\phi_i(\mathbf{x}'), \tag{2.16}$$

where $\phi_1,\dots,\phi_n : \mathbb{R} \to \mathbb{R}$ are simple known functions [Wilson et al., 2020a, Rasmussen and Williams, 2006]. Thus, a zero-mean Gaussian process[1] with such kernel $f \sim \mathcal{GP}(0,k)$ can be approximated as a linear transformation of a standard multivariate Gaussian variable $\varepsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I})$

$$f(\cdot) = \sum_{i=1}^{n} \phi_i(\cdot)\varepsilon_i. \tag{2.17}$$

The key to efficient sampling is to combine these two approximations, which we can do with the following formula[2]

$$\underbrace{f(\cdot)|\mathbf{U}}_{\text{sparse posterior}} \overset{d}{=} \underbrace{f(\cdot)}_{\text{prior}} + \underbrace{k(\cdot,\mathbf{Z})k(\mathbf{Z},\mathbf{Z})^{-1}(\mathbf{U}-f(\mathbf{Z}))}_{\text{update}}, \tag{2.18}$$

[1] We take a zero-mean Gaussian process for clarity in equations. The formula easily applies to Gaussian processes with non-zero mean.

[2] This is a generalisation of Matheron's rule to sparse Gaussian processes.

illustrated in Figure 2.3.

Thus, combining Equation (2.18) with the approximation of the prior in Equation (2.17), we obtain

$$f(\cdot)|\mathbf{U} \approx \sum_{n=1}^{N} \phi_n(\cdot)\varepsilon_n + k(\cdot,\mathbf{Z})k(\mathbf{Z},\mathbf{Z})^{-1}\left(\mathbf{U} - \sum_{n=1}^{N} \phi_n(\mathbf{Z})\varepsilon_n\right). \tag{2.19}$$

By sampling $\varepsilon \leftarrow \mathcal{N}(\mathbf{0},\mathbf{I})$ and $\mathbf{U}$ from the variational posterior, we can sample a function from the variational posterior in $O\left(n+m^2n+m^3\right) = O\left(m^2n+m^3\right)$ time **?**.

2.3 Deep Gaussian Processes

Deep Gaussian processes are a multi-layer generalization of Gaussian processes - that is, a deep Gaussian process F with L layers is defined as a composition of L Gaussian processes $(f_1,\ldots,f_L)$

$$F = f_L \circ \cdots \circ f_1. \tag{2.20}$$

The hierarchical structure of deep Gaussian processes makes them more expressive than shallow Gaussian processes, somewhat similarly to how neural networks are more expressive than general linear models. However, due to this composite structure we also lose some of the elegant properties of Gaussian processes. For instance, inference in deep Gaussian processes is challenging because the posterior is no longer tractable, and approximations must be employed. Several frameworks for approximate inference in deep Gaussian processes have been proposed, including the variational inference approach of Damianou and Lawrence [2013], the approximate expectation propagation method of Bui et al. [2016], and the doubly-stochastic variational inference framework introduced by Salimbeni and Deisenroth [2017] which we focus on in this work.

In the doubly-stochastic variational inference framework, the intractable Gaussian process posterior is approximated with a variational posterior, where effectively the Gaussian process at each layer is replaced with a sparse Gaussian process introduced in Section 2.2. This way, the variational posterior is simply the product of the variational posteriors at each layer

$$q(\mathbf{F}^l,\mathbf{U}^{l\,L}_{l=1}|\mathbf{Z}^L_{l=1}) = \prod_{l=1}^{L} p(\mathbf{F}^l|\mathbf{U}^l;\mathbf{F}^{l-1},\mathbf{Z}^{l-1})q(\mathbf{U}^l|\mathbf{Z}^{l-1}), \tag{2.21}$$

where $\mathbf{Z}^l,\mathbf{U}^l$ are the inducing locations and inducing values at the l-th layer, and $\mathbf{F}^l$ is the output of the l-th layer, with $\mathbf{F}^0 = \mathbf{X}$ by definition. Because of this factorised form, one will not be surprised to find that the equation for the ELBO finds an analogous form to one for the shallow sparse Gaussian process

$$\text{ELBO}_{\text{deep}} = \sum_{i=1}^{n} \mathbb{E}_{q(\mathbf{F}_i^L|\mathbf{U}^L)}[\log p(\mathbf{y}_i|\mathbf{F}_i^L)] - \sum_{l=1}^{L} \text{KL}(q(\mathbf{U}^l)||p(\mathbf{U}^l|\mathbf{Z}^{l-1})). \tag{2.22}$$

Moreover, because the marginal posterior $q(\mathbf{F}_i^l)$ of a sparse shallow Gaussian process depends only its input $\mathbf{F}_i^{l-1}$, we obtain samples from $q(\mathbf{F}_i^L)$ via hierarchical sampling of $\mathbf{F}_i^1,;\mathbf{F}_i^2|\mathbf{F}_i^1,;\cdots,;\mathbf{F}_i^L|\mathbf{F}_i^{L-1}$ for an efficient approximation of the expectation in

Equation (2.22). Additionally, the layered factorised structure lends itself to efficient sampling of functions from the deep Gaussian process posterior via pathwise sampling described in Section 2.2.2, using an analogous hierarchical approach, though this time sampling from the full posteriors rather than the marginals.

? propose one more trick to improve the performance of deep Gaussian processes in the doubly stochastic variational inference framework. Namely, they suggest using a linear mean function for the Gaussian processes at the hidden layers, partly inspired by the residual connections in ResNets He et al. [2015]. In particular, when the input and output dimensions of a hidden layer are equal, they define

$$\mu(\mathbf{x}) = \mathbf{x}. \tag{2.23}$$

This allows the deep Gaussian process to model residuals between layers rather than the full function. In effect, rather than starting from scratch, each layer adds its own gradual contribution to the representation attained at the previous layer.

This gradual process fits the layered structure well and seems to make the inference with deep Gaussian processes significantly easier.

2.4 Gaussian Processes on Manifolds

Although any Euclidean kernel k can be used with data on a Riemannian manifold X when it is embedded in some Euclidean space, there are reasons to look for a more appropriate kernel when data pertains to a non-Euclidean manifold. For instance, one likely desires for the covariance of the values of a Gaussian process at two different points to be proportional to the distance between them. However, the distance on a manifold may be significantly different from the Euclidean distance in its embedding space. Additionally, some spaces may exhibit symmetries that the Euclidean space does not exhibit, which we want the kernel to respect - also the reverse might be true where the Euclidean kernel respects symmetries not inherently present in the manifold. Indeed, the difference between a Euclidean kernel on an embedding and a properly defined kernel on a manifold can be seen in Figure 2.4 for the case of a ”broken ring” manifold.

Recent pioneering work [Lindgren et al., 2011, Borovitskiy et al., 2020] has generalised the Matern class of kernels to a wide class of Riemannian manifolds. The generalisation, however, is not as simple as replacing the Euclidean distance in Equation (2.9) with the geodesic distance (natural distance on a particular manifold), as the resulting kernel is not generally positive semi-definite (see Equation (2.7)) [Costa et al., 2023]. Borovitskiy et al. [2020] showed that a Matern kernel properly generalised to a Riemannian manifold X can be expressed as the infinite series

$$k_{\nu,\kappa,\sigma^2}(\mathbf{x},\mathbf{x}') = \frac{\sigma^2}{C_{\nu,\kappa}} \sum_{n=0}^{\infty} \Phi(\lambda_n)\phi_n(\mathbf{x})\phi_n(\mathbf{x}') \tag{2.24}$$

$$\Phi(\lambda_n) = \begin{cases} \left(\frac{2\nu}{\kappa^2} + \lambda_n\right)^{-\nu-\frac{d}{2}} & \text{if } \nu \in (0,\infty) \\ e^{-\frac{\kappa^2}{2}\lambda_n} & \text{if } \nu = \infty \end{cases}$$

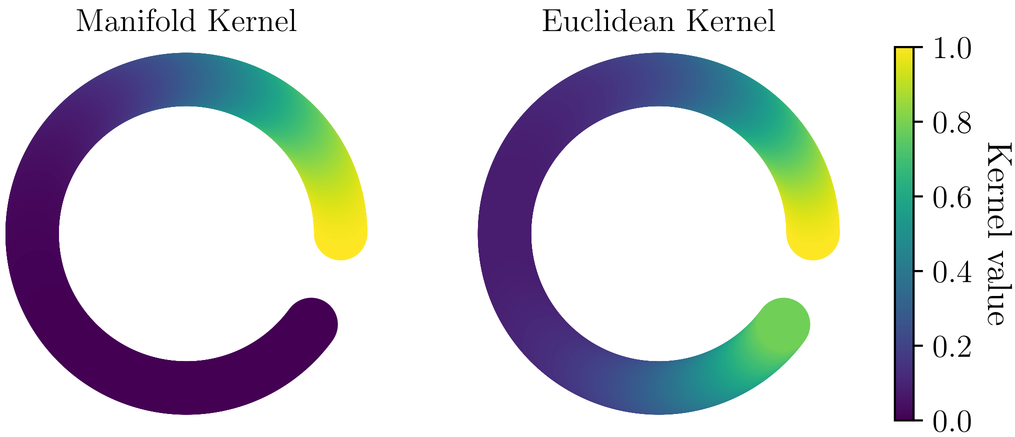

Figure 2.4: Comparison of values of the Euclidean RBF kernel (see Equation (2.9)) and a geometry-aware RBF kernel (see Equation (2.24)) on a manifold resembling a broken ring. This is properly a 1-dimensional manifold, with thickness added for clearer visualisation. The color at any point of the manifold represents the kernel value between that point and the top end of the broken ring. Because the Euclidean kernel is a function of the Euclidean distance between points, it gives a high value between the two ends of the broken ring. The manifold kernel respects the geodesic distance inherent to the manifold and its value between these points is close to zero.

where $\{\phi_n\}_{n\geq 0}$ are eigenfunctions of the Laplace-Beltrami operator Δ on X with respective eigenvalues $\{\lambda_n\}_{n\geq 0}$, d is the dimension of X, and $C_{\nu,\kappa}$ is a normalization constant. In practice, this series is truncated for computational tractability, which is theoretically justified through guarantees on the speed of convergence of the series in Equation (2.24) [Azangulov et al., 2022].

Since our prior discussion on variational inference in Gaussian processes was kernel-agnostic, shallow Gaussian processes on Riemannian manifolds defined using the kernel in Equation (2.24) enjoy the benefits of efficient inference and training via inducing points (the inducing locations must be on the manifold!). Additionally, we can see that this kernel has the form given in Equation (2.17), which allows for efficient sampling of functions from sparse Gaussian process posteriors on Riemannian manifolds built using this kernel via pathwise sampling (Section 2.2.2).

Notice that we did not draw the corresponding conclusions for deep Gaussian processes, as the outputs of Gaussian processes on manifolds do not necessarily lie on the input manifold. We show how this issue is addressed with residual deep Gaussian processes in the next chapter.

Chapter 3

Residual Deep Gaussian Processes on Manifolds

In this chapter, we detail the construction of residual deep Gaussian processes on Riemannian manifolds and evaluate their performance in modelling irregular functions with a synthetic regression task on the sphere. We examine the performance of our model as data density increases and demonstrate that, when data is abundant enough, residual deep Gaussian processes, owing to their layered structure, can offer improved performance over their shallow counterparts in regression with irregular target functions.

Additionally, we introduce a construction of deep Gaussian processes on Riemannian manifolds based on a geometry-aware input layer composed with a Euclidean deep Gaussian process. This model serves as a baseline against which we evaluate residual deep Gaussian processes. We demonstrate that, while both our model and this baseline offer an advantage over shallow Gaussian processes, our model achieves superior median performance across all depths tested and exhibits less variance in performance across independent experiments. We find, however, that in individual experiment runs the baseline performance can be higher than that of our model.

3.1 Model Construction

Residual deep Gaussian processes generalise deep Gaussian processes to Riemannian manifolds. They attempt to utilise the expressive power of the layered structure of deep Gaussian processes while respecting the geometric structure of the manifold, by modelling each layer as a manifold-to-manifold ”Gaussian” process[1].

Residual deep Gaussian processes address the generally non-trivial issue of constructing manifold-to-manifold Gaussian processes by modelling displacement vectors of the input points with a Gaussian vector field. Such a vector field can be used to translate the input points on the manifold when composed with the *exponential map*, thereby

[1] We put ”Gaussian” in quotes, since, strictly speaking, a manifold-output random variable cannot be Gaussian in the usual sense, because Gaussian distributions are vector-valued [Mallasto and Feragen, 2018].

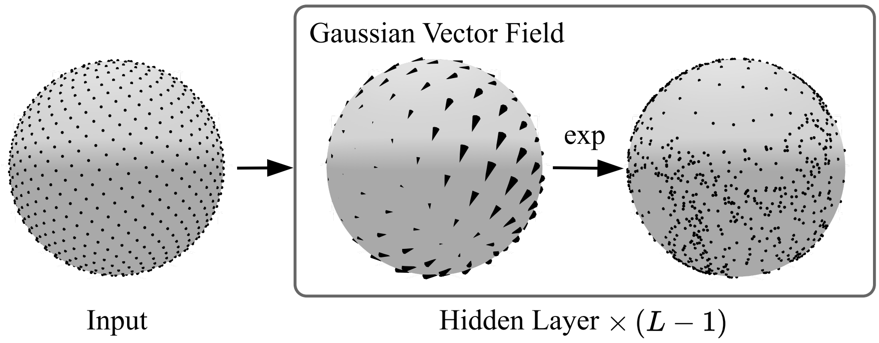

Figure 3.1: Schematic illustration of a residual deep Gaussian process on the sphere without the final layer. *Left:* Input points uniformly distributed over the sphere. *Middle:* A sample from a Gaussian vector field modelling translation vectors by which to translate the input points. *Right:* The input points, previously in a uniform grid, after translation by the sample vector field via the exponential map. Residual deep Gaussian processes achieve expressive representations by repeated translations of the input via a composition of Gaussian vector fields with the exponential map.

creating a distribution over manifold-to-manifold functions, which we call a manifold-to-manifold Gaussian process. Roughly speaking, each Gaussian vector field models the difference between the output distribution and the input, or the *residual* — hence the name of the model.

In the following, we describe two ways of building Gaussian vector fields in a computationally viable way, which we utilise in the experiments in Section 3.2, Chapter 4, and Chapter 5. We also briefly detail the recently proposed intrinsic construction of Gaussian vector fields, which we highlight as a promising direction for future applications of residual deep Gaussian processes.

3.1.1 Gaussian Vector Fields on Manifolds

Gaussian vector fields are a special case of vector-valued Gaussian processes, which take values in the space tangent to their domain. They have been shown to be useful modelling tools for many tasks, such as modelling wind velocities on the globe Robert-Nicoud et al. [2023] or learning equations of motion in a physical system Hutchinson et al. [2021].

Specifically, if f is a Gaussian vector field on a d-dimensional manifold X, then $f(\mathbf{x})$ is a random vector normally distributed in the space tangent to X at $\mathbf{x}$, denoted by $\mathcal{T}_{\mathbf{x}}X$. In Euclidean space, Gaussian vector fields are simply vector-valued Gaussian processes discussed in Section 2.1 and can be implemented, for instance, by stacking independent Gaussian processes into a vector. With non-Euclidean manifolds, the construction cannot be as simple, since, for example, on the sphere $\mathbb{S}^2$ embedded in $\mathbb{R}^3$ a sample from a vector-valued Gaussian process constructed by naively stacking independent Gaussian processes will generally not be tangential to the sphere.

Several solutions to this problem have been proposed, including constraining a vector-

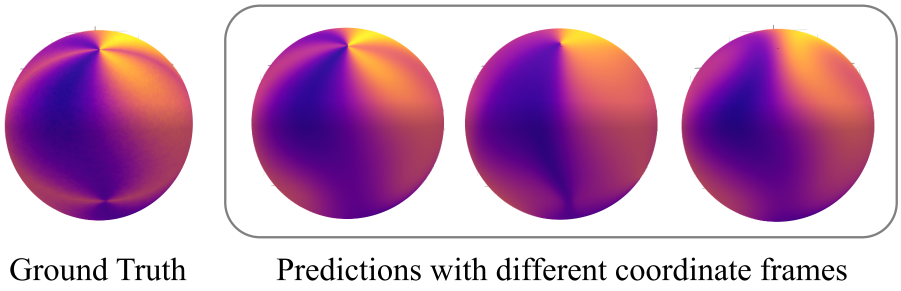

Figure 3.2: Comparison of the posterior mean of a residual deep Gaussian process on the sphere based on the coordinate frame construction of Gaussian vector fields for different choices of coordinate frames. The objective function is shown on the left, while three predictions, of quality varying with the choice of the coordinate frame, are shown on the right.

valued Gaussian process to lie in the tangent space of a manifold embedded in Euclidean space through linear constraints [Lange-Hegermann, 2018]. In our work, however, we focus on the computationally simple methods proposed by Hutchinson et al. [2021] that integrate seamlessly with the framework of sparse Gaussian processes we have presented thus far.

Hutchinson et al. [2021] introduced two computationally viable ways to construct Gaussian vector fields on manifolds:

1. **Projection of embedded Gaussian processes**: Given an isometric embedding $\text{emb}: X \to \mathbb{R}^n$ and a vector-valued Gaussian process $\mathbf{g}$ on $\text{emb}(X)$, a Gaussian vector field $\mathbf{f}$ on X can be defined as

$$\mathbf{f}(\mathbf{x}) = \mathbf{P}_{\mathbf{x}}\mathbf{g}(\text{emb}(\mathbf{x})) \tag{3.1}$$

   where $\mathbf{P}_{\mathbf{x}}$ is a position-dependent orthogonal projection from $\mathbb{R}^d$ onto the tangent space $T_{\text{emb}(\mathbf{x})}X$ with a known simple form.[2]

2. **Coordinate frame in tangent spaces**: Given a coordinate frame (i.e., a system of vector fields spanning the tangent space at every point) $(\mathbf{e}_1, \dots, \mathbf{e}_d)$ on X and a vector-valued Gaussian process $\mathbf{g} = (\mathbf{g}_1, \dots, \mathbf{g}_d)$ on X, a Gaussian vector field $\mathbf{f}$ can be defined as $\mathbf{f}(\mathbf{x}) = \sum_{i=1}^{d} \mathbf{g}_i(\mathbf{x})\mathbf{e}_i(\mathbf{x})$.

Theoretically, any Gaussian vector field on X can be produced through either of these methods. Nevertheless, the choice between the two constructions turns out to be quite subtle. For example, different choices of coordinate frames can result in a drastically different quality of predictions, as illustrated in Figure 3.2.

Very recently, an intrinsic construction of Gaussian vector fields was proposed, which addresses some practical problems of the previous constructions — for example, its uncertainty is strictly non-increasing with distance, which is not necessarily the case for the two constructions above [Robert-Nicoud et al., 2023]. Similarly to how scalar-valued Matern kernels on manifolds were built from eigenfunctions of the Laplace–Beltrami

[2] $\mathbf{P}_{\mathbf{x}}$ is the adjoint of the differential $\text{d}_{\mathbf{x}}\,\text{emb}$.

operator, Robert-Nicoud et al. define these Gaussian processes using a matrix-valued Matern kernel built from *eigenfields* of the Hodge Laplacian Δ. Specifically, the Matern kernel defining an intrinsic Gaussian vector field is given by

$$k_{\nu,\kappa,\sigma^2}(\mathbf{x},\mathbf{x}') = \frac{\sigma^2}{C_{\nu,\kappa}} \sum_{n=0}^{\infty} \Phi_{\nu,\kappa}(\lambda_n)\mathbf{s}_n(\mathbf{x}) \otimes \mathbf{s}_n(\mathbf{x}'), \tag{3.2}$$

where $\{\mathbf{s}_n\}_{n\geq 0}$ are the eigenfields of Δ with respective eigenvalues $\{\lambda_n\}_{n\geq 0}$, $\otimes$ denotes the tensor product, and $\Phi_{\nu,\kappa}$ is a simple function related to the kernel parameters.

This construction offers fundamental advantages over the previous two solutions. For instance, it turns out that the eigenfields $\{\mathbf{s}_n\}_{n\geq 0}$ may be chosen such that each $\mathbf{s}_n$ is either *curl-free*, *divergence-free*, or *harmonic*. Intuitively, curl-free fields exhibit no vortices, while divergence-free fields have no sinks or sources (harmonic fields are not relevant to our work, since no harmonic eigenfields exist on hyperspheres). Consequently, by constructing a kernel using the eigenfields from only one of these classes, we can obtain a distribution supported on that class of vector fields [Robert-Nicoud et al., 2023]. This can be useful in certain problems, for instance in modelling wind velocity at certain altitudes, where a divergence-free bias could follow from the physics of the modelled processes **?**.

At the time of writing, the intrinsic construction is only computationally viable for the hypertori $\mathbb{T}^d$, the sphere $\mathbb{S}^2$, and arbitrary products thereof. Nevertheless, it remains a promising direction for future applications of residual deep Gaussian processes for tasks such as modelling of wind velocity close to the Earth's surface, where the complexity of the turbulent wind behaviour may be modelled better by a residual deep Gaussian process than a shallow Gaussian process.

Because each of the constructions of Gaussian vector fields given above can be built either *from* sparse Gaussian processes (the projected and coordinate-frame constructions) or *as* a sparse Gaussian process (the intrinsic construction), all three constructions enjoy the benefits of efficient variational inference and pathwise sampling. These properties will be transferred to residual deep Gaussian processes.

3.1.2 Residual Deep Gaussian Processes

Residual deep Gaussian processes are a generalisation of Euclidean deep Gaussian processes whose every layer is a manifold-to-manifold Gaussian process. Concretely, on a Riemannian manifold X, a residual deep Gaussian process F is a repeated composition of a sequence of Gaussian vector fields $f, \ldots, f^{L-1} : X \to \mathcal{T}X$ alternating with the exponential map $\exp_{\mathbf{x}} : \mathcal{T}_{\mathbf{x}}X \to X$ and a final manifold Gaussian process g.

$$F = g \circ (\exp \circ f^{L-1}) \circ \cdots \circ (\exp \circ f^1). \tag{3.3}$$

g can be scalar-valued or vector-valued depending on the modelling task.

As we have mentioned before, each Gaussian vector field f^l models a distribution over displacements of its inputs f^{l-1}, which is applied to f^{l-1} via the exponential map. Thus, intuitively speaking, each layer models only a small increment to the representation

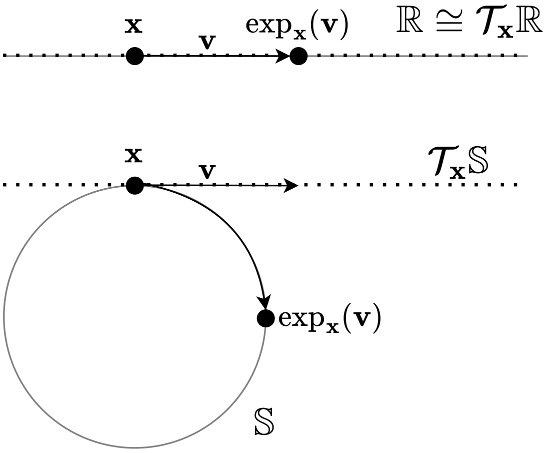
$\mathbf{x}$
$\mathbf{v}$
$\exp_{\mathbf{x}}(\mathbf{v})$
$\mathbb{R} \cong \mathcal{T}_{\mathbf{x}}\mathbb{R}$
$\mathbf{x}$
$\mathbf{v}$
$\mathcal{T}_{\mathbf{x}}\mathbb{S}$
$\exp_{\mathbf{x}}(\mathbf{v})$
$\mathbb{S}$

Figure 3.3: Schematic illustration of the difference between tangent spaces and exponential maps on the 1-dimensional unit sphere $\mathbb{S}$ (bottom) and the 1-dimensional Euclidean space $\mathbb{R}$ (top). Because the tangent space at any point in $\mathbb{R}$ can be identified with $\mathbb{R}$ itself, and thus with the tangent space at any other point, the exponential map can be identified with vector addition i.e. $\exp_{\mathbf{x}}(\mathbf{v}) = \mathbf{x} + \mathbf{v}$. This is not true for the 1-sphere $\mathbb{S}$ as the above identifications cannot be made due to the curvature of $\mathbb{S}$.

attained at the previous layer. This is a natural scheme for models with layered structure bearing similarities to successful models in deep learning such as the ResNet [He et al., 2015].

One may recall that we made precisely the same arguments for the Euclidean deep Gaussian processes with a linear mean (see Equation (2.23)). This is no coincidence, as residual deep Gaussian processes turn out to be a generalisation of these models. Indeed, in a Euclidean space $\mathbb{R}^d$ the tangent space $\mathcal{T}\mathbf{x}\mathbb{R}^d$ can be identified with the tangent space at any other point $\mathbf{x}'$ and with $\mathbb{R}^d$ itself. Consequently, the exponential map can be seen simply as vector addition $\exp\mathbf{x}(\mathbf{x}') = \mathbf{x} + \mathbf{x}'$. Thus, in $\mathbb{R}^d$, a zero-mean Gaussian vector field composed with the exponential map is equivalent to a vector-valued Gaussian process with the linear mean defined in Equation (2.23)

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x},\mathbf{x})) = \mathbf{x} + \mathcal{GP}(0, k(\mathbf{x},\mathbf{x})). \tag{3.4}$$

Finally, as we have mentioned in Section 3.1.1, because the Gaussian vector fields we described can be constructed from sparse Gaussian processes, the doubly-stochastic variational inference framework can be applied directly to residual deep Gaussian processes. Indeed, using this framework, our model enjoys efficient training and inference, as well as efficient sampling of functions from its posterior via pathwise sampling (see Section 2.2.2), which we utilise in Chapter 4 for approximation of acquisition functions.

A Simple Baseline Residual deep Gaussian processes are not the only way to generalise deep Gaussian processes to Riemannian manifolds. In fact, one could construct a deep Gaussian process F on a manifold X as a composition of a shallow geometry-aware Gaussian process $f : X \to \mathbb{R}^n$ and a Euclidean deep Gaussian process $F_{\text{Euclidean}} : \mathbb{R}^n \to \mathbb{R}^m$

$$F = F_{\text{Euclidean}} \circ f. \tag{3.5}$$

We created this model to serve as a simple baseline for comparison with residual deep Gaussian processes in Section 3.2.

We hypothesised that, owing to their layered structure, residual deep Gaussian processes are effectively more expressive than shallow Gaussian processes on manifolds. Thus, we expected our model to achieve superior performance in tasks that require increased expressiveness, such as regression on an irregular function.

3.2 Evaluation on Stylised Examples

To test our hypothesis that residual deep Gaussian processes are more expressive in practice than shallow geometry-aware Gaussian process, we constructed an irregular function on the sphere, shown in Figure 3.2 as ground truth. It was designed so that in multiple regions the function's partial derivatives increase rapidly and, in fact, explode to infinity at several locations. We evaluated how well residual deep Gaussian processes can model this function as their depth increases and as the number of training points changes. We also compared the performance of our model against the baseline model defined in Equation (3.5).

Methodology We evaluated the residual deep Gaussian process model with 1, 2, 3, 4, and 5 layers, where a single-layer model is a shallow (geometry-aware) Gaussian process. To avoid the nuances in choosing a coordinate frame, which could drastically impact performance (see Figure 3.2), we used the projected construction of Gaussian vector fields described in Section 3.1.1. We trained our model for 1000 steps using the Adam optimiser [Kingma and Ba, 2017] on 100, 200, 400, and 800 training points distributed in a uniform grid on the sphere. We evaluated our model on 2000 points distributed uniformly on the sphere collecting the log predictive density (LPD), test log likelihood (TLL), and mean squared error (MSE). Given a posterior distribution p and a test set of inputs $\mathbf{X} \in X^{d \times n}$ and labels $\mathbf{y} \in \mathbb{R}^n$ these metrics are defined as follows:

$$\text{LPD} = \log p(\mathbf{y}|\mathbf{X}) \tag{3.6}$$

$$\text{TLL} = \sum_{i=1}^{n} \log p(\mathbf{y}_i|\mathbf{X}_i) \tag{3.7}$$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left|\left|\mathbf{y}_i - \mathbb{E}_{y \sim p(\cdot|\mathbf{X}_i)}[y]\right|\right|^2. \tag{3.8}$$

Each experiment was repeated 5 times to obtain an estimate of variance in performance due to stochasticity in training and inference.

Because the posterior p of a residual deep Gaussian process is generally intractable, we cannot simply compute these metrics using the above equations. We circumvent this

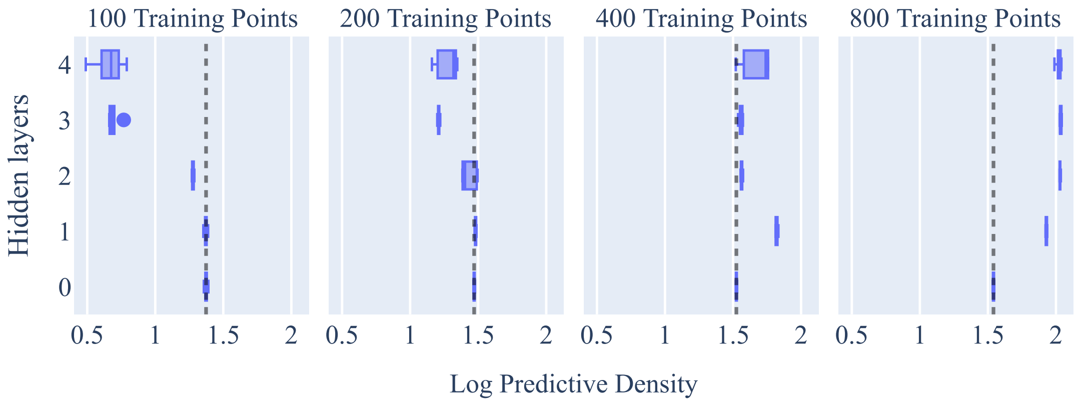

Figure 3.4: Performance of residual deep Gaussian processes in modelling an irregular target function on the sphere in terms of log predictive density (more to the right is better). The number of hidden layers of the model is shown on the y-axis with 0 indicating that the tested model was a classical sparse Gaussian process. Boxes show the interquartile range of the test set performance over the 5 independent runs. The horizontal line inside each box indicates the median performance. Whiskers extend to the observations furthest away from the median, but still within 1.5 times the interquartile range from either the lower or upper quartiles, while dots represent the outliers beyond that range. The vertical dotted grey lines show the median performance of the shallow Gaussian processes.

issue, taking for instance LPD, by applying the following manipulation

$$\begin{aligned} \mathrm{LPD} &= \log p(\mathbf{y}|\mathbf{X}) && (3.9) \\ &= \log \mathbb{E}_{\mathbf{F}^{L-1}\sim p(\cdot|\mathbf{X})}\left[p(\mathbf{y}|\mathbf{F}^{L-1},\mathbf{X})\right] && (3.10) \\ &= \log \mathbb{E}_{\mathbf{F}^{L-1}\sim p(\cdot|\mathbf{X})}\left[p(\mathbf{y}|\mathbf{F}^{L-1})\right], && (3.11) \end{aligned}$$

where $\mathbf{F}^{L-1}$ is the output of the $(L-1)$-th layer of the residual deep Gaussian process corresponding to the test inputs $\mathbf{X}$. The expectation in Equation (3.11) can be approximated with the Monte Carlo method by pathwise sampling (see Section 2.2.2) through the first $L-1$ layers of the residual deep Gaussian process. Indeed, an analogous manipulation can be applied to obtain sampling-based approximations of TLL and MSE for deep Gaussian processes. We should mention that, because the TLL and MSE factorise over the test points, they can also be approximated by sampling from marginal distributions.

We use the model defined in Equation (3.5) as a baseline for comparison to residual deep Gaussian processes in the densest data regime of 800 training points. In contrast to residual deep Gaussian processes, the performance of the baseline model varies significantly between experiments. To improve the robustness of our analysis considering this variation, we increased the number of experiment repetitions to 15 for the 800 training points setting.

Results Figure 3.4 illustrates the performance of residual deep Gaussian processes on our stylised regression task in terms of log predictive density. We find that our model

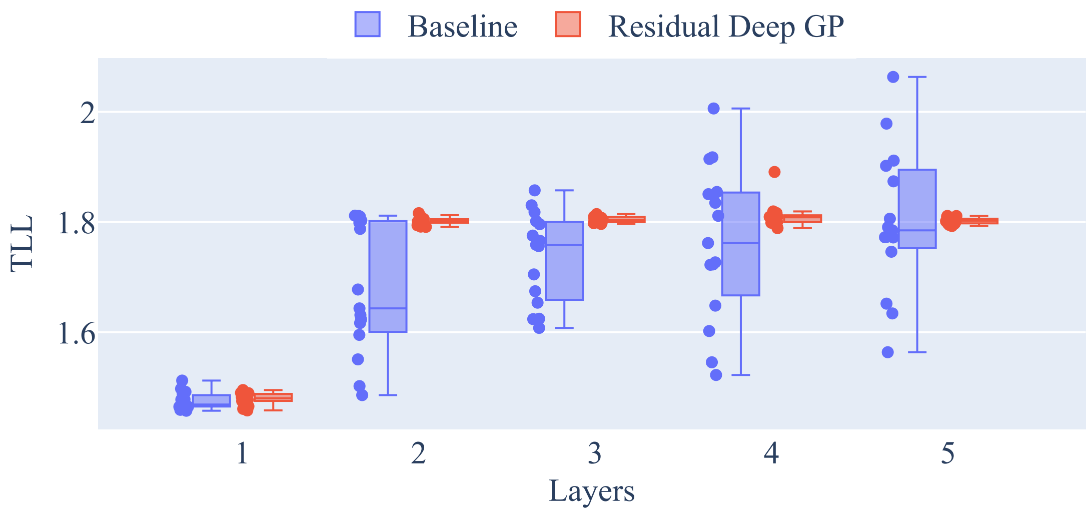

Figure 3.5: Performance of residual deep Gaussian processes in modelling an irregular target function on the sphere compared against the baseline model defined by Equation (3.5). Y-axis indicates the test log-likelihood (see Equation (3.7)). Boxes show the interquartile range of the test-set performance over the 5 independent runs. The horizontal line inside each box indicates the median performance. Whiskers extend to the observations furthest away from the median but still within 1.5 times the interquartile range from either the lower or upper quartiles, while dots represent the performance for each individual run.

with 2 layers and above is outperformed in the sparse data regimes of 100 and 200 training points by the shallow Gaussian process. In principle, residual deep Gaussian processes should be able to recover the performance of a model with no hidden layers by setting the output-scale (see Section 2.1) of the Gaussian vector fields to 0. Our experiments show that, although theoretically possible, this may not happen in practice. Especially as the depth increases, the probability of recovering the performance of shallow models seems to decrease, and worse solutions are produced.

With 400 training points, there is a clear improvement in the performance of residual deep Gaussian processes. The model with one hidden layer is able to fit the true function significantly better than the shallow model, while exhibiting performance consistent between runs. Models with 2 and 3 hidden layers perform marginally better than shallow Gaussian processes. Interestingly, the median performance of the deepest model is equal to approximately 1.75 — much closer to the 2-layered model (≈ 1.80) than to the shallow model (≈ 1.50). Thus, we observe that the regime of 400 points seems to be a threshold temporarily destabilising the trend of performance, depth, and data density.

Indeed, with 800 training points, we find that the trend stabilises again in a more intuitive pattern inverting the relation of performance to depth seen in the regime of 100 training points. More concretely, in the densest data regime, the median log predictive density of residual deep Gaussian processes increases with depth, saturating with 3 hidden layers, at a value around 2.03. The deepest model achieves similar log predictive density to the model with 3 hidden layers, though it exhibits slightly more variance in its performance across train-test runs, which is an expected effect, as training of deeper models is generally less stable than the training of shallower ones [He et al., 2015]. Nevertheless, from our data it appears that residual deep Gaussian processes become more stable between independent experiments as the number of training points increases. This appears logical, since with a higher number of observations the variability in functions that fit the data and have a high enough prior probability should diminish.

In Figure 3.5, we show a comparison of the performance of residual deep Gaussian processes to the baseline performance of the geometry-aware Gaussian process composed with a Euclidean deep Gaussian process. We see that although both models can be considered generalisations of deep Gaussian processes to manifolds, their performance in terms of test log-likelihood is substantially different — both in terms of the median and across individual runs.

We can notice that residual deep Gaussian processes perform much more consistently than the baseline model between runs and their median test log-likelihood is higher than that of the baseline across all layers. In fact, the median performance of the baseline model, across all choices of the number of layers, is lower than the lowest median performance of the (non-shallow) residual deep Gaussian process. It is worth mentioning that the test log-likelihood of the residual deep Gaussian process saturates already at 1 hidden layer, whereas the more comprehensive metric, log predictive density, increases gradually until 3 hidden layers are used. This suggests that there is an improvement in the fit of residual deep Gaussian processes when increasing the depth beyond 1 hidden layer that is not captured by a metric based on the marginal posterior distribution.

Although generally worse in the median performance, the baseline performance does exceed that of our model on some individual runs — to a larger extent as the depth of the baseline model increases. One reason for this could be optimisation of the inducing locations (see Section 2.2). In our setup, residual deep Gaussian processes on the sphere use a uniform grid of points as the inducing locations which are fixed after initialisation. On the other hand, the baseline model optimises inducing locations for all layers, initialising all locations via (non-deterministic) k-means clustering of the training data, as done in Salimbeni and Deisenroth [2017].

Optimising inducing locations is a high-dimensional problem, which, combined with the non-deterministic initial state, can make model performance inconsistent. It is possible that some exceptionally well-performing arrangements of inducing points are found with the help of randomness in certain runs, while in others, the opposite is true. Nonetheless, further investigation could help determine what is the driving factor behind the exceptionally good, and the exceptionally bad, test-set performance of the baseline in some runs. Furthermore, pathwise sampling could be implemented for Euclidean space, to enable computation of log predictive density of the baseline model, giving a more comprehensive picture of its performance.

Key Findings We have demonstrated, through regression on a stylised function on the sphere, that residual deep Gaussian processes can significantly outperform shallow Gaussian processes in modelling irregular functions when enough data is available. In fact, in the densest data regime tested, we found that model performance increases monotonically with model depth until it stagnates at some value. We also noticed that this trend is roughly reversed when data is sparse, diminishing model performance with increased depth, despite a theoretical ability to recover the shallow Gaussian process solutions by the residual deep Gaussian processes, likely due to the increased complexity of optimising a deeper model.

Additionally, we found that the baseline model constructed by composing a geometry-aware input Gaussian process with a Euclidean deep Gaussian process did outperform our model in certain individual experiment runs. Nevertheless, residual deep Gaussian processes, even with only one hidden layer, outperformed all tested configurations of the baseline model in terms of median test log-likelihood. Finally, we pointed out the remarkable stability between experiment runs that our model exhibits, suggesting that robustness to stochasticity in training may be an inherent strength of residual deep Gaussian processes.

Chapter 4

Bayesian Optimisation of Irregular Functions

In this chapter, we investigate whether residual deep Gaussian processes can be successfully applied to optimise functions on hyperspheres via Bayesian optimisation. We focus on functions irregular around their optima, which shallow Gaussian processes may struggle to model due to their simplicity bias. We find that residual deep Gaussian processes can offer an advantage for such irregular functions, especially when combined with shallow Gaussian processes in a strategic manner.

The chapter is structured as follows: first, we introduce Bayesian optimization and discuss geometry-aware Bayesian optimization; then, we delve into deep Gaussian processes for Bayesian optimization, including the adaptation of acquisition functions; finally, we present our experiments on synthetic benchmarks and discuss the results, demonstrating the effectiveness of residual deep Gaussian processes for optimizing irregular functions on hyperspheres.

4.1 Bayesian Optimisation

Bayesian optimization is a powerful technique for data-efficient optimization of black-box functions. It is most often applied where functions of interest are expensive to evaluate, whether in terms of time, capital, or computational resources. For these reasons, Bayesian optimization has found success in automatic optimization of deep learning architectures (where training and evaluating large models is both computationally and temporally expensive) Masum et al. [2021], drug discovery (where the desired function of drug effectiveness requires extensive testing or simulation to evaluate) Pyzer-Knapp [2018], as well as aerospace engineering and robotics (where evaluation of component design or actuator settings requires time-consuming simulations) Lam et al. [2018], Jaquier et al. [2021].

The key idea of Bayesian optimisation is to construct a probabilistic model of the function we want to optimise $y : X \to \mathbb{R}$, typically using a Gaussian process. We then use this model iteratively to decide where to evaluate y, updating the model with the new

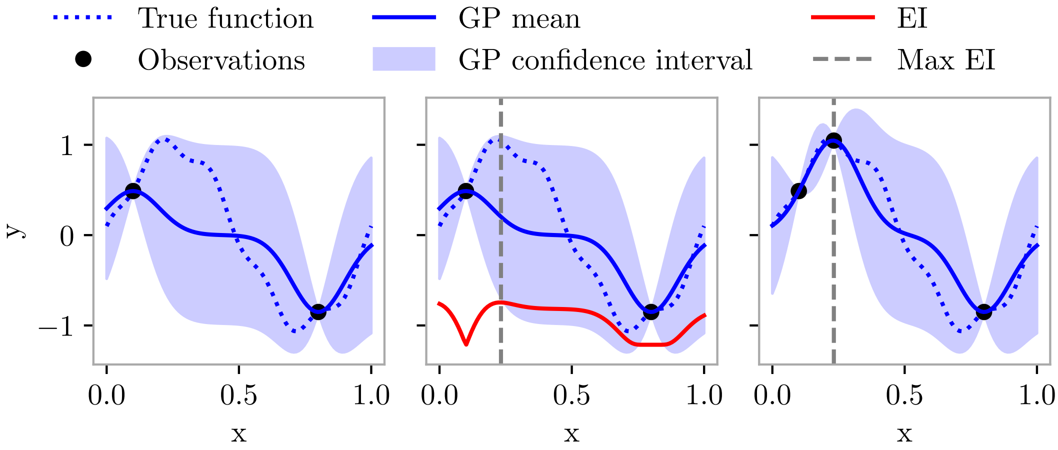

Figure 4.1: An illustration of a Bayesian optimisation step. *Left:* A Gaussian process fitted to two observations acquired thus far. *Middle:* A fitted Gaussian process and the corresponding expected improvement (EI) acquisition function with maximum indicated by a vertical line. *Right:* A Gaussian process fitted to three observations with the newest observation acquired at the maximum point of the expected improvement.

observation before performing the decision again. Specifically, starting with a set of initial observations $\mathcal{D}_n = (\mathbf{x}_i, y_i)_{i=1}^n$, where $y_i = y(\mathbf{x}_i)$, Bayesian optimization repeatedly performs the following steps until a stopping criterion is met, such as a maximum number of iterations or a convergence threshold (see Figure 4.1 for an illustration):

1. **Fit a Gaussian process $f \sim \mathcal{GP}(\mu, k)$ to the observations $\mathcal{D}$.** Such f quantifies a belief about the target function via its posterior mean and covariance. A low posterior variance at a point indicates that enough data has been collected to be quite sure about the value of the objective function at that point, whereas a high posterior variance suggests that the nearby region of the search space has not been explored enough to confidently say whether the value of y there is low or high.

2. **Define an acquisition function $a : X \to \mathbb{R}$.** This function quantifies how advantageous observing the value of y would be to the optimization process, with higher values indicating more advantageous choices. Acquisition functions [Frazier, 2018] leverage the probabilistic information from the posterior Gaussian process $f(\cdot)|\mathcal{D}$ to balance exploration of the unknown search space regions and exploitation of the already acquired knowledge.

3. **Find a point $\mathbf{x}_{n+1}$ maximising a and observe y at that point.** Typically such maximum needs to be approximated with a chosen optimisation method. If X exhibits a geometric structure, geometry-aware optimisation methods can be employed to utilise this structure.

4. **Add the latest observation $y_{n+1} = y(\mathbf{x}_{n+1})$ to the set of all observations $\mathcal{D}_{n+1} = \mathcal{D}_n \cup (\mathbf{x}_{n+1}, y_{n+1})$.**

If the search space X exhibits a geometric structure, the optimisation process can be enhanced by taking this geometry into account. In many tasks, such as geostatics

[Hutchinson et al., 2021] or robotics [Jaquier and Rozo, 2020], data is inherent to a non-Euclidean manifold X. The geometric structure of the manifold can be utilised to improve the Bayesian optimisation process. This can be done using a Gaussian process properly defined on X that respects the domain's symmetries, geodesic distance, etc. Additionally, geometric information can be incorporated into the process of finding the maximum of acquisition functions. Indeed, exploiting the geometric structure of the search space with these two methods in conjunction has been shown to make the Bayesian optimisation process more efficient [Jaquier and Rozo, 2020, Jaquier et al., 2021].

4.1.1 Deep Gaussian Processes for Bayesian Optimization

We can notice in the outline of Bayesian optimisation we provided that no assumptions about the form of the objective function are made. This means that, in principle, Bayesian optimisation can be used with arbitrarily complex objective functions. Nevertheless, some assumptions about the objective function need to be made, in this case through the choice of the mean function μ and covariance function k defining the Gaussian process f. Indeed, if no assumptions at all can be made, then every optimisation problem is equally likely, in which case, the No Free Lunch theorem tells us that, in expectation, we can do no better than a random search [Wolpert and Macready, 1997]. Yet, if the objective function is highly irregular around the optimum, the commonly used Gaussian processes, in particular Matérn Gaussian processes, may be unable to accurately model that region of the search space due to their bias towards simpler, slowly varying functions. This, in turn, would impede the optimisation from discovering the optima.

To address this issue, we can utilise deep Gaussian processes, which are more expressive than shallow Gaussian processes in the sense that the distributions over functions they define are less biased towards simple functions [Salimbeni and Deisenroth, 2017]. Indeed, the layered structure of deep Gaussian processes often allows them to fit irregular functions more accurately than shallow Gaussian processes, as seen in **??**.

When utilising deep Gaussian processes for Bayesian optimization, the computation of acquisition functions needs to be adapted. Specifically, many acquisition functions that have a closed form expression when paired with a shallow Gaussian process, are no longer tractable when using a deep Gaussian process (or a residual deep Gaussian process). For instance, one of the most commonly used acquisition functions in Bayesian optimisation, and one that we use in Section 4.2, is the expected improvement

$$a(\mathbf{x}) = \mathbb{E}_{y \sim f(\mathbf{x})}\left[\max(0, y - y^*)\right], \qquad f(\mathbf{x}) \sim \mathcal{N}\left(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x})\right), \tag{4.1}$$

where y^* is the best observed value thus far and $\max(0, y - y^*)$ is the *improvement*. We can notice that the improvement term is non-negative, which can be seen as a manifestation of the heuristic *optimism in the face of uncertainty* - we take into account only the potential gain hoping for a large improvement even if the point considered also has a probability of being a much worse solution than the best one found thus far.

Unfortunately, for a deep Gaussian process the expected improvement does not generally have a closed form. Nevertheless, we can efficiently approximate this acquisition function by approximating Equation (4.1) with the Monte Carlo method. Specifically, since the expectation depends only on the marginal posterior $f(\mathbf{x})$, we can approximate the expected value with an average over a large but finite number of samples. These can be obtained using either hierarchical sampling as discussed in Section 2.3. In fact, in Section 2.3 we have seen that the marginal distribution of a deep Gaussian process at a point depends only on the location of that point and on no other test points. Thus, it is easy to efficiently sample points from the marginal posterior through hierarchical sampling and thus we can approximate the expectation in Equation (4.1) with the Monte Carlo method.

Moreover, we can make a more general statement using pathwise sampling. Indeed, through pathwise sampling we can apply deep Gaussian processes, and residual deep Gaussian processes, with any acquisition function that relies either on samples from the posterior (here we are not restricted to the marginal posterior), in which case we can use it directly, or to evaluate expectations over posterior distributions, in which case we can approximate them with the Monte Carlo method.

Indeed, we shall see that these approximations can be successful, as in the next section we apply residual deep Gaussian processes with the expected improvement acquisition function approximated via pathwise sampling.

4.2 Evaluation of Synthetic Benchmarks

To test whether residual deep Gaussian processes can help improve Bayesian optimization for irregular functions, we tested our model on two functions: the Ackley function projected on $\mathbb{S}^3$ for direct comparison to Jaquier et al. [2021] and a custom function with a singularity near the optimum on $\mathbb{S}^2$, the exact definitions of these functions are given in Appendix A.

Methodology We tested our model and the baseline shallow Gaussian process in a Bayesian optimization process with 200 acquisition steps for each of the target functions. In the case of residual deep Gaussian processes, we performed the first 180 Bayesian optimization steps with an exact Matérn Gaussian process and followed that with 20 steps using a residual deep Gaussian process. This was motivated by our previous findings in Section 3.2, where we found that residual deep Gaussian processes could make significant improvements over shallow Gaussian processes, but only where data is abundant enough. Thus, the optimization starts in a sparse data regime where shallow Gaussian processes seem to perform better and collects data around the optimum to form a sufficiently dense regime where a residual deep Gaussian process finds a considerably better fit. In Figure 4.2, we reported the logarithm of the regret attained by the Bayesian optimization process across the 200 optimization steps, which is defined simply as

$$\text{regret}(y_{\text{candidate}}, y_{\text{optimum}}) = |y_{\text{candidate}} - y_{\text{optimum}}| \tag{4.2}$$

where $y_{\text{candidate}}$ is the best function value found thus far, and y_{optimum} is the global optimum.

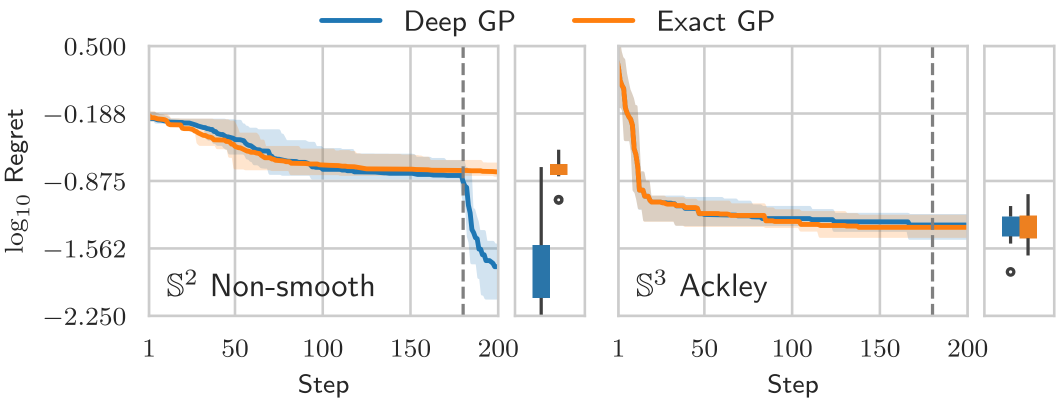

Figure 4.2: *Left:* Logarithm of the difference between the true optimum and the optimum discovered via Bayesian optimization. *Right:* Custom function with an irregular behavior around the optimum. *Right:* The Ackley function, which is quite smooth around the optimum. The blue line shows the regret when the first 180 acquisition steps are done with an exact Gaussian process followed by 20 steps with a residual deep Gaussian process, while the orange line corresponds to optimization that relies solely on a shallow Gaussian process.

For both models, we used the expected improvement acquisition function, which we approximate using pathwise sampling for the deep model. It should be noted that when the acquisition function is being optimized, the deep Gaussian process posterior will be evaluated multiple times. In order for this optimization to be done consistently, one needs to ensure that the expected improvement is approximated using the samples from the Gaussian process posterior rather than resampling between evaluations of the acquisition function. Because the initialization and optimization process is stochastic to a large extent, we repeated each experiment 15 times to obtain an estimate of the effect of the random seed used.

Our implementation of residual deep Gaussian processes is integrated with GPyTorch [Gardner et al., 2018], which allowed us to use the BoTorch library [Balandat et al., 2020] for implementing Bayesian optimization. Additionally, following Jaquier et al. [2021], we used Pymanopt [Townsend et al., 2016] for geometry-aware gradient-based optimization of acquisition functions. All parameters that were unmentioned here are constant and fixed between experiments and models. We give their values in Appendix A.

Results Figure 4.2 compares the experimental performance of both methods across one example of an irregular function (right) and one example of a regular function (left), showing the trajectory of the median log regret as well as the interquartile range in the log regret exhibited by individual runs.

We found that when optimizing the function on the sphere with a singularity near the optimum, switching to a residual deep Gaussian process in the latter stage of the optimization process results in significant, and often immediate, improvement in the quality of the optimum found. The shaded regions indicating interquartile range show that this happens to a significant degree in almost all cases. However, looking at the box plot of the final regret, we can see that there are outliers to this trend - more specifically, we found one outlier out of 15 Bayesian optimization runs that did not result in an improvement. We have observed that sometimes, due to the randomness in the initialization and optimization of the acquisition function, the data that was close enough to the singularity was not acquired in the initial 180 steps to make the irregular objective function likely, even given the expressive prior distribution of a residual deep Gaussian process. In these cases, the deep Gaussian process posterior looked similar to the shallow Gaussian process posterior, and the region around the true optimum was not explored further. Nevertheless, in the vast majority of cases, the shallow Gaussian process did collect some data close enough to the optimum to ”suggest” a complex function, which, though it could not fit due to its simplicity, the residual deep Gaussian process could.

Looking at the results for the Ackley function projected on $\mathbb{S}^3$, we find practically no noticeable difference in performance between the two methods. Both median trajectories look almost exactly the same and reproduce the results in Jaquier et al. [2021]. This is precisely what we had suspected, as the region around the minimum of this objective function is smooth. We expected that a shallow Gaussian process would be able to fit that region of space almost perfectly and thus the residual deep Gaussian process would offer no advantage in terms of the quality of its fit to the data. We do see one outlier in favor of the residual deep Gaussian process; however, we see similar occasionally better-performing runs for the Bayesian optimization of the irregular function using only a shallow Gaussian process. Due to a large amount of randomness in Bayesian optimization in general, we can likely attribute it to random variation rather than an advantage of our model in optimizing smooth functions.

Our results provide evidence that deep residual Gaussian processes offer an advantage over shallow Gaussian processes when modeling irregular functions on manifolds. Moreover, since our model excels when data is abundant, using shallow Gaussian processes to gather an initial set of data before switching to a deep residual Gaussian process is an effective strategy when the initial data set is small, as is usually the case in common Bayesian optimization pipelines. Following this intuition, we expect that more elaborate strategies may be used to further exploit this symbiotic relation. For instance, periodically or randomly switching between the two models would be a generalization of our strategy and could prove effective in optimizing irregular functions with multiple local optima. This would bear resemblance to a heuristic based on randomly switching between acquisition functions, which has seen some practical success [Noskova and Borovitskiy, 2023].

Key findings To summarize, we demonstrated that residual deep Gaussian processes can significantly improve the performance of Bayesian optimization for irregular functions on hyperspheres. Specifically, we showed that a Bayesian optimization procedure on hyperspheres that began with a shallow Gaussian process but finished with a residual deep Gaussian process can avoid the difficulty of fitting residual deep Gaussian processes to sparse datasets seen in Section 3.2 while benefiting from their greater

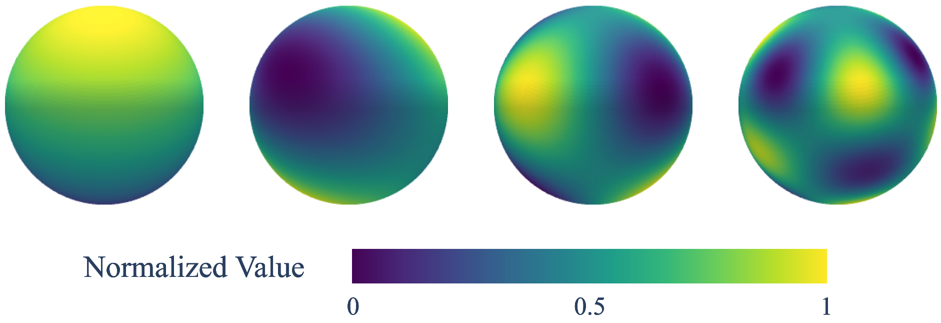

Figure 4.3: Examples of spherical harmonics on the sphere. The frequency of the harmonics increases moving between subplots from left to right.

expressiveness when the optimization process stagnates. Indeed, we indicated that a more adaptive scheme - for instance, one that switches between the models when no improvement is made for a fixed number of optimization steps - would exploit the symbiotic relationship between deep and shallow models for Bayesian optimization to a higher degree.

We also observed that for functions that behave smoothly around their optima, such as the Ackley function projected on $\mathbb{S}^3$, there was little to no benefit from switching to a residual deep Gaussian process toward the end of the optimization process. This finding suggests that the increased expressiveness of deep models may not always be necessary, and that the choice between shallow and deep Gaussian processes should be informed by the characteristics of the objective function, such as its regularity.

Chapter 5

Approximate Inference for Euclidean Data

In this chapter, we examined whether residual deep Gaussian processes on hyperspheres can serve as an alternative to Euclidean deep Gaussian processes for inference tasks with data in the Euclidean space. We motivate this investigation by the recent work of Dutordoir et al. [2020], who showed that by projecting Euclidean data onto a high-dimensional sphere and applying a geometry-aware Gaussian process on this hypersphere, one can achieve good prediction performance while obtaining a significant gain in inference speed over Euclidean models. The key driver behind this strategy is a novel class of variational Gaussian processes proposed by Dutordoir et al. [2020]. These leverage spherical harmonics — functions defined on hyperspheres analogous to sinusoids in Euclidean spaces — for increased inference speed and obtaining a global approximation of the posterior. Crucially, because spherical harmonics are only defined on hyperspheres, a projection is necessary if the data is not initially located on a hypersphere.

Because each layer of a residual deep Gaussian process on a hypersphere takes its values in a hypersphere, each layer can utilise the variational approximation proposed by Dutordoir et al. [2020]. This makes residual deep Gaussian processes an appealing candidate for adapting this class of variational Gaussian processes to a multi-layer architecture.

To test whether residual deep Gaussian processes can improve upon standard Gaussian processes with this variational inference strategy, we extend the experiments of Dutordoir et al. [2020] and benchmark our model on several datasets from the *UCI Machine Learning Repository* [Kelly et al., 2023]. We also compare the performance of our model to Euclidean deep Gaussian processes applied directly to the data without projection onto a hypersphere. We find that, on most datasets, residual deep Gaussian processes offer superior performance than shallow Gaussian processes with the variational approximation based on spherical harmonics. Our comparison to Euclidean deep Gaussian processes requires additional experiments, as our results do not correspond well to the ones reported in [Salimbeni and Deisenroth, 2017].

Before moving on to the experiments, we briefly detail the variational Gaussian process construction based on spherical harmonic features and describe why it justifies the choice of projecting Euclidean data onto a hypersphere.

5.1 Variational Inference with Spherical Harmonic Features

In Section 2.2.1, we have seen that a Gaussian process $f \sim \mathcal{GP}(\mu, k)$ can be conditioned not only on (potentially noisy) observations of f, but also on observations of $\mathrm{T} f$, where T is a linear transformation $T : L_2(X_1) \to L_2(X_2)$. In fact, we saw that a Gaussian process f conditioned on observations of $\mathrm{T} f$ is another Gaussian process whose mean and covariance functions are defined by simple closed-form expressions.

Dutordoir et al. [2020] utilise this property to construct a particularly well-behaved family of variational Gaussian processes on hyperspheres $\mathbb{S}^d$. Specifically, they use the linear transformation

$$\mathrm{T} : f \mapsto \langle f, \cdot \rangle \tag{5.1}$$

and a fixed family of inducing points $\mathbf{Z}$ in the transformed domain, where $\mathbf{Z}_i$ is the i-th spherical harmonic ϕ_i on $\mathbb{S}^d$ (see Figure 4.3 for an illustration). Thus, the value $\mathbf{U}_i$ at $\mathbf{Z}_i$ is defined as

$$\mathbf{U}_i = \langle f, \phi_i \rangle. \tag{5.2}$$

Because spherical harmonics form an orthonormal basis of $L_2(\mathbb{S}^d)$, f satisfies the identity

$$f = \sum_{i=1}^{\infty} \langle f, \phi_i \rangle \phi_i, \tag{5.3}$$

where each coefficient $\langle f, \phi_i \rangle$ gives us some information about the global behaviour of f. This is akin to how Fourier coefficients give us information about the global behaviour of periodic functions.

Recalling our discussion in Section 2.2, variational inference with inducing points $\mathbf{Z}$ tries to approximate the true posterior $p(\mathbf{U}|\mathbf{Y};\mathbf{Z},\mathbf{X})$ with a variational distribution $q(\mathbf{U}|\mathbf{Z}) = \mathcal{N}(\mathbf{m}, \mathbf{S})$. Intuitively, with spherical harmonics playing the role of inducing points, we are trying to estimate the distribution of the coefficients of spherical harmonics in a function drawn from the posterior Gaussian process. Indeed, because each coefficient $\langle f, \phi_i \rangle$ tells us something about the global behaviour of the Gaussian process, optimising these variational coefficients amounts to naturally approximating the posterior in a global way, rather than trying to summarise the posterior by describing its behaviour at a set of concrete input locations.

Additionally, because the coefficients $\langle f, \phi_i \rangle$, $\langle f, \phi_j \rangle$ have zero covariance when $i \neq j$, the covariance matrix $k(\mathbf{Z}, \mathbf{Z})$ and thus the matrix inversion in Equation (2.12) can be performed in linear, rather than cubic, time. Consequently, the asymptotic complexity of computing the variational posterior reduces to $O\left(m^2 b\right)$ compared to $O\left(m^2 b + m^3\right)$ for the standard inducing points approximation, where b is the batch size. Indeed, Dutordoir et al. claim that this asymptotic improvement is a significant reason for the computational acceleration seen in practice.[1]

Because this method can only be applied to Gaussian processes on hyperspheres, a projection is needed when data is not initially on a hypersphere. Dutordoir et al. propose to use the following projection

$$\Psi : \mathbb{R}^d \to \mathbb{S}^d \subseteq \mathbb{R}^{d+1}; \quad \mathbf{x} \mapsto \frac{\mathbf{x}_b}{||\mathbf{x}_b||}, \tag{5.4}$$

where $(\mathbf{x}_b)_i = \mathbf{x}_i$ for $1 \leq i \leq d$ and $(\mathbf{x}_b)_i = b \in \mathbb{R}$ for $i = d+1$, where b is a *bias* term that offsets the hyperplane $\mathbb{R}^d$ embedded in $\mathbb{R}^{d+1}$ away from the origin. Additionally, y is scaled down by a factor of $||\mathbf{x}_b||$ at training; though, at test time, the target values are not scaled down, and the posterior distribution is rescaled appropriately. This choice of projection does, in fact, have a theoretical justification based on works on Gaussian processes as limits of neural networks of infinite size [Cho and Saul, 2009, Rasmussen and Williams, 2006].

Following its inventors, we at times refer to the variational approximation described above as the spherical harmonic features approximation. Indeed, residual deep Gaussian processes on hyperspheres lend themselves well to the spherical harmonic features approximation, as the projection Equation (5.4) need only be applied before the input layer and not thereafter, since hidden layers also take their values on hyperspheres.

5.2 Evaluation on UCI datasets

Methodology We evaluate residual deep Gaussian processes on the *Kin8nm*, *Power*, *Concrete*, *Boston*, *Yacht*, and *Energy* regression datasets from the UCI machine learning repository [Kelly et al., 2023] using the spherical harmonic features variational approximation at each layer and compare them against Euclidean deep Gaussian processes. We train our model with 1, 2, 3, 4, and 5 layers, where a model with 1 layer is simply a shallow Gaussian process, and report the performance in terms of test log likelihood (see Equation (3.7)). Our experimental setup follows that of Salimbeni and Deisenroth [2017], except that we scale the number of training epochs down from 10000 to 1000, since after 1000 epochs, we saw little to no performance improvements in the tested models. We train our models using the Adam optimiser [Kingma and Ba, 2017] on a randomly selected training set containing 90% of the data and test each model on the remaining 10% of the dataset, repeating each experiment 5 times. Following Dutordoir et al. [2020], we use 210 inducing variables (spherical harmonics for residual deep Gaussian processes; inducing points for Euclidean deep Gaussian processes) for the Concrete and Kin8nm datasets, 336 for the Power dataset, 294 for the Yacht dataset, and 119 for the Boston dataset.

We used the code provided by the authors of Dutordoir et al. [2020] for computation of spherical harmonics in our implementation of the corresponding variational inference

[1] It is not clear whether this is actually caused by the asymptotic improvements, as typically the batch size would indeed be larger than the number of inducing points. Nevertheless, this method yields multiple simplifications hidden by the asymptotic notations, which surely contribute to the improvement of inference speed.

technique. For the initial projection of datasets from $\mathbb{R}^d$ to $\mathbb{S}^d$, we used equation *Equation* (5.4) with $b = 1$. We motivated this choice by the fact that for larger values of b, the distances between projected data points would be lower than for $b = 1$ and would thus require higher frequency harmonics to accurately model, while for $b < 1$, data could lie inside the sphere before projection, making it an intuitively doubtful choice. We attempted optimisation of b from training data; however, such optimisation hindered the test-set performance as it quickly led to overfitting.

All variables that we have not mentioned here were held constant, and their values are reported in Appendix A.

Results In Figure 5.1, we show the performance of residual deep Gaussian processes across the number of layers on the six datasets we considered, compared against the performance of Euclidean deep Gaussian processes.

Focusing, for now, on the performance of residual deep Gaussian processes only, we find that it improves with depths for the Kin8nm, Power, and Concrete datasets. Indeed, the greatest increase in performance can be seen on Kin8nm and Power, which have the highest ratio of the number of data points to the dimension of the dataset. This correlation between performance of deep models and data density is not at all unexpected. As we have seen in Section 3.2, data density appears to be a crucial factor in determining whether performance can be gained from increasing the depth of residual deep Gaussian Processes. With 1030 data points, Concrete is the third largest out of the tested datasets; thus, an improvement in performance with depth is not unexpected, and the fact that the improvement does not appear as dramatic as in the case of Kin8nm seems logical, since the latter dataset has 8 times as many data points. We should stress the word ”appear” in the previous sentence, since it is hard to quantify the rate of improvement across datasets. Indeed, concrete conclusions may only be drawn from the relative values of test log-likelihoods within one dataset.

Comparing the performance of our model to the performance of Euclidean deep Gaussian processes, we find that the Euclidean models boast superior performance, until a rapid collapse at 4 and 5 layers, except for the Power dataset. It is difficult to say why this collapse happens, since our results for 1-3 layers (except for the Boston dataset) approximately match the values reported by Salimbeni and Deisenroth [2017] on the same datasets then adjusted for the lower number of training epochs. It is only at higher number of layers that our results differ in the performance pattern exhibited — through, again, this difference is not seen for the Power dataset.

Looking for differences between our model and the Euclidean models, we hypothesised that this collapse might be caused by numerical instabilities due to inducing locations being too close to each other. This could be due either to a number of inducing points that was too high or optimisation of inducing locations incidentally drawing them close together. Nevertheless, we repeated the experiments controlling for both variables and continued to find the same performance patterns (see Appendix B). The fact that the model appears best behaved on the largest dataset could point to issues related to dataset size; however, further experiments need to be conducted to gain insight into this seemingly anomalous behaviour. For this reason, we direct most of our attention away from the Euclidean models in these experiments.

For the Boston and Energy datasets, which are smaller than the other three we discussed but still larger than the Yacht dataset, we find that residual deep Gaussian processes offer little to no advantage over shallow models. This aligns with our expectations of the gains from depth being limited by data density. Interestingly, we find that in the Boston dataset, our 3-layered model performs noticeably and consistently worse than the others. It is not clear why this particular setting repeatedly exhibits a poorer fit to the data; however, this can be interpreted as an instance of a general trend. Specifically, both in the performance patterns on UCI datasets and in the synthetic data in Section 3.2, we have seen that the variance in solutions found by residual deep Gaussian processes for problems on hyperspheres appears significantly lower than that of Euclidean models or manifold-Euclidean hybrids. If true in general, this would make our model potentially more reliable than its Euclidean counterpart when the effect of a stochastic training process is considered.

Finally, we see that for the Yacht dataset, the performance of our model deteriorates with depths, the performance of a shallow Gaussian process is recovered by the model with one hidden layer. This situation is extremely similar to the performance pattern we saw for the regime of 200 or 100 training points on the sphere in Section 3.2 and with little surprise as Yacht is the smallest out of the six datasets with 278 training points in 6 dimensions.

Key Findings To summarise, we successfully reproduced the results of Dutordoir et al. [2020] for regression with shallow Gaussian processes based on spherical harmonic features on the tested UCI datasets and showed that their performance can be enhanced with increasing depth of residual deep Gaussian processes on the three datasets with the largest data density. We found that on the smallest dataset, Yacht, the performance deteriorates with depth, while on the two datasets with a slightly higher data point count, increasing depth results in almost no changes to performance.

Finally, we also noticed that here, as in the stylised example in Section 3.2, the variance in performance of residual deep Gaussian processes is noticeably smaller than that of their fully or partially Euclidean counterparts. This could mean that our model boasts robustness to stochasticity in training.

Throughout, we commented on the seemingly anomalous performance of Euclidean deep Gaussian processes, concluding that a further investigation is needed, as our initial tests seemed to rule out numerical issues due to optimisation of inducing locations.

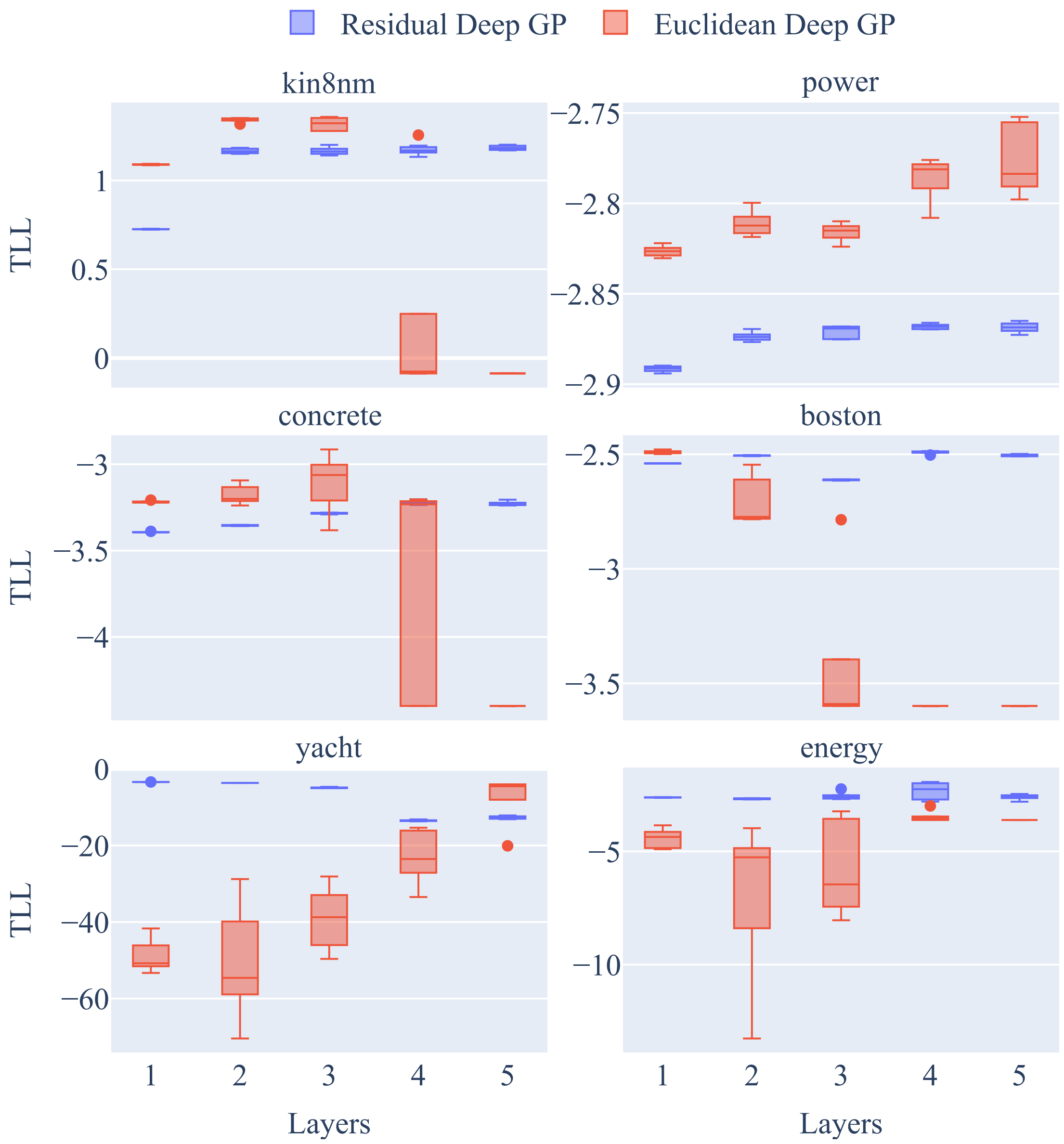

Figure 5.1: Performance of deep residual Gaussian processes and Euclidean deep Gaussian processes on six datasets from the UCI machine learning repository. The number of layers in each model is given on the x-axis, with 1 indicating a shallow Gaussian process. Boxes show the interquartile range of the test-set performance over the 5 independent runs. The horizontal line inside each box indicates the median performance. Whiskers extend to the observations furthest away from the median but still within 1.5 times the interquartile range from either the lower or upper quartiles, while dots represent the outliers beyond that range.

Chapter 6

Conclusion and Future Work

Gaussian processes have proven to be a powerful probabilistic tool for learning unknown functions, particularly in tasks such as Bayesian optimization, active learning, and reinforcement learning where accurate uncertainty estimates are crucial. Data in these tasks often has inherent geometric structure - in particular, it may reside on a non-Euclidean manifold. A range of Gaussian process constructions has been developed to harness the geometry of data for improved predictions; however, these methods are nevertheless often limited by the simplicity bias of their defining kernels. Deep Gaussian processes hope to alleviate this limitation by sequentially combining multiple Gaussian processes; however, thus far, little has been known about their generalisation to Riemannian manifolds — Residual deep Gaussian processes.

In this fourth year project, we conducted a comprehensive investigation of residual deep Gaussian processes, aiming to develop a multi-faceted understanding of their strengths and weaknesses. We hypothesised that these models would offer improved performance in modelling irregular functions and may even offer competitive performance in regression on Euclidean data by projecting it onto a hypersphere. To this end, we conducted three broad experiments, while focusing our scope to hypersphere domains. Firstly, we conducted a fundamental analysis of our model in a fully synthetic experimental setting, evaluating the joint impact of model depth and data density on the performance of residual deep Gaussian processes using a target function specially designed to test the model's ability to capture irregular functions on hyperspheres. Secondly, we explored the application of residual deep Gaussian processes to one of the most prominent use cases for Gaussian processes: Bayesian optimization. Lastly, we evaluated our model on the ambitious task of modeling Euclidean data by projecting it onto a hypersphere and employing a specialized variational inference strategy based on spherical harmonics.

Our experiments on modeling an irregular function on the sphere demonstrated that residual deep Gaussian processes can significantly outperform shallow Gaussian processes when sufficient training data is available, supporting our main hypothesis. We observed a monotonic increase in model performance with increasing depth, saturating at around 3 hidden layers. Notably, our model exhibited superior median performance compared to a simplified deep Gaussian process baseline across all depths tested, while also showing remarkable stability across independent experimental runs. However, we also found that in sparse data regimes, the performance of residual deep Gaussian processes deteriorates with increasing depth, despite their theoretical ability to recover the solutions of shallow Gaussian processes. This behavior is likely due to the increased complexity of optimizing deeper models.

Leveraging the insights gained from the controlled experiments, we proceeded to showcase the effectiveness of residual deep Gaussian processes in optimizing irregular functions on hyperspheres via Bayesian optimization. By strategically combining shallow Gaussian processes for initial exploration with residual deep Gaussian processes for subsequent exploitation, we achieved significant improvements in the quality of the discovered optima. This symbiotic approach harnesses the strengths of both models, mitigating the challenges faced by deep models in sparse data regimes. Moreover, we confirmed that for functions that behave smoothly around their optima, residual deep Gaussian processes offer no significant advantage over standard Gaussian processes.

In an ambitious attempt to extend the applicability of our model to Euclidean data, we explored the projection of real-world datasets onto hyperspheres and employed a variational approximation based on spherical harmonics. Our experiments on several datasets from the UCI Machine Learning Repository demonstrated improved performance over shallow Gaussian processes, with the performance gains being most pronounced on datasets with higher data density. These findings align with our observations from the controlled synthetic experiments and suggest that manifold learning techniques can enhance the applicability and performance of residual deep Gaussian processes when dealing with Euclidean data. Interestingly, we also noticed that the variance in performance of residual deep Gaussian processes was noticeably smaller than that of their fully or partially Euclidean counterparts, suggesting that our model may be more robust to stochasticity in training.

While our model achieved performance close to Euclidean deep Gaussian processes on these tasks, the comparison was hindered by anomalous behavior exhibited by the Euclidean models in our experiments. Further investigation is needed to determine the factors behind this inconsistency and to establish a more definitive comparison.

Looking ahead, our findings open up several promising avenues for future research. The success of the adaptive strategy for Bayesian optimization suggests that more elaborate schemes, such as periodically switching between shallow and deep models, could further enhance performance on irregular functions with multiple local optima. Additionally, the recently introduced intrinsic Gaussian vector field construction presents an exciting opportunity to apply residual deep Gaussian processes to complex real-world problems, such as modeling wind velocities near the Earth's surface.

By demonstrating the effectiveness of residual deep Gaussian processes in regression and Bayesian optimization tasks, identifying their strengths and weaknesses in various data regimes, and exploring their application to Euclidean data, we have contributed to the growing body of knowledge on these expressive models. As research in this field continues to evolve, we anticipate that residual deep Gaussian processes will play an increasingly important role in tackling challenging problems involving data on non-Euclidean manifolds.

Bibliography

Mathieu Alain, So Takao, Brooks Paige, and Marc Peter Deisenroth. Gaussian processes on cellular complexes. 2023.

Iskander Azangulov, Andrei Smolensky, Alexander Terenin, and Viacheslav Borovitskiy. Stationary kernels and gaussian processes on lie groups and their homogeneous spaces i: the compact case. 2022.

Iskander Azangulov, Andrei Smolensky, Alexander Terenin, and Viacheslav Borovitskiy. Stationary kernels and gaussian processes on lie groups and their homogeneous spaces ii: non-compact symmetric spaces. 2023.

Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A framework for efficient monte-carlo bayesian optimization, 2020.

Edwin V Bonilla, Kian Chai, and Christopher Williams. Multi-task gaussian process prediction. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2007.

Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. Matérn gaussian processes on riemannian manifolds. 2020.

Thang Bui, Daniel Hernandez-Lobato, Jose Hernandez-Lobato, Yingzhen Li, and Richard Turner. Deep gaussian processes for regression using approximate expectation propagation. In *Proceedings of The 33rd International Conference on Machine Learning*. PMLR, 2016.

Youngmin Cho and Lawrence Saul. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2009.

Nathael Da Costa, Cyrus Mostajeran, Juan-Pablo Ortega, and Salem Said. Invariant kernels on riemannian symmetric spaces: a harmonic-analytic approach, 2023.

Sam Coveney, Cesare Corrado, Caroline H. Roney, Daniel O'Hare, Steven E. Williams, Mark D. O'Neill, Steven A. Niederer, Richard H. Clayton, Jeremy E. Oakley, and Richard D. Wilkinson. Gaussian process manifold interpolation for probabilistic atrial activation maps and uncertain conduction velocity. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2020.

Andreas C. Damianou and Neil D. Lawrence. Deep gaussian processes, 2013.

Marc Peter Deisenroth and Carl Edward Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011.

Vincent Dutordoir, Nicolas Durrande, and James Hensman. Sparse gaussian processes with spherical harmonic features. 2020.

David Duvenaud, James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani. Structure discovery in nonparametric regression through compositional kernel search, 2013.

Stefanos Eleftheriadis, Dominic Richards, and James Hensman. Sparse gaussian processes with spherical harmonic features revisited. 2023.

Bernardo Fichera, Viacheslav Borovitskiy, Andreas Krause, and Aude Billard. Implicit manifold gaussian process regression. 2023.

Peter I. Frazier. A tutorial on bayesian optimization. 2018.

Jacob R. Gardner, Geoff Pleiss, David Bindel, Kilian Q. Weinberger, and Andrew Gordon Wilson. GPyTorch: Blackbox matrix-matrix gaussian process inference with GPU acceleration, 2018.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

James Hensman, Nicolo Fusi, and Neil D. Lawrence. Gaussian processes for big data, 2013.

James Hensman, Nicolas Durrande, and Arno Solin. Variational fourier features for gaussian processes, 2017.

Michael Hutchinson, Alexander Terenin, Viacheslav Borovitskiy, So Takao, Yee Whye Teh, and Marc Peter Deisenroth. Vector-valued gaussian processes on riemannian manifolds via gauge independent projected kernels. 2021.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. 2018.

Noémie Jaquier and Leonel Rozo. High-dimensional bayesian optimization via nested riemannian manifolds. 2020.

Noémie Jaquier, Viacheslav Borovitskiy, Andrei Smolensky, Alexander Terenin, Tamim Asfour, and Leonel Rozo. Geometry-aware bayesian optimization in robotics using riemannian matérn kernels. 2021.

Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The uci machine learning repository. `https://archive.ics.uci.edu`, 2023.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Rémi Lam, Matthias Poloczek, Peter Frazier, and Karen E. Willcox. *Advances in Bayesian Optimization with Applications in Aerospace Engineering*. 2018.

Markus Lange-Hegermann. Algorithmic linearly constrained gaussian processes, 2018.

Finn Lindgren, Håvard Rue, and Johan Lindstr”om. An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2011.

Anton Mallasto and Aasa Feragen. Wrapped gaussian process regression on riemannian manifolds. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Mohammad Masum, Hossain Shahriar, Hisham Haddad, Md Jobair Hossain Faruk, Maria Valero, Md Abdullah Khan, Mohammad A. Rahman, Muhaiminul I. Adnan, Alfredo Cuzzocrea, and Fan Wu. Bayesian hyperparameter optimization for deep neural network-based network intrusion detection. In *2021 IEEE International Conference on Big Data (Big Data)*, 2021.

Ekaterina Noskova and Viacheslav Borovitskiy. Bayesian optimization for demographic inference. *G3: Genes, Genomes, Genetics*, 2023.

E. O. Pyzer-Knapp. Bayesian optimization for accelerated drug discovery. *IBM Journal of Research and Development*, 2018.

Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 2005. URL `http://jmlr.org/papers/v6/quinonero-candela05a.html`.

Carl Edward. Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

Daniel Robert-Nicoud, Andreas Krause, and Viacheslav Borovitskiy. Intrinsic gaussian vector fields on manifolds. 2023.

Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep gaussian processes. 2017.

Matthias W. Seeger, Christopher K. I. Williams, and Neil D. Lawrence. Fast forward selection to speed up sparse gaussian process regression. PMLR, 2003.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms, 2012.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 2012.

James Townsend, Niklas Koep, and Sebastian Weichwald. Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation, 2016.

Mark van der Wilk, Vincent Dutordoir, ST John, Artem Artemev, Vincent Adam, and

James Hensman. A framework for interdomain and multioutput gaussian processes, 2020.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning, 2015.

James T. Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. Efficiently sampling functions from gaussian process posteriors. 2020a.

James T. Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. Pathwise conditioning of gaussian processes. 2020b.

D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1997.

Appendix A

Experimental Details

A.1 Regression on the Sphere

For each model we used the following hyperparameters

- Matern kernel with an initial length-scale of 1, output-scale of 1, and smoothness of 2.5. All hyperparameters were fruther learned during optimisation.
- An output-scale Gamma prior with.
- 60 inducing points initialised uniformly on the sphere.

A.2 Bayesian Optimisation of Irregular Functions

We used the following Bayesian optimisation parameters

- An initial of 5 observations sampled uniformly at random from the hypersphere.
- The Log-Expected Improvement acquisition function.

The acquisition function was optimised using the Riemannian *steepest descent* optimisation algorithm implemented in Pymanopt [Townsend et al., 2016]. After each acquisition step we retrained the Gaussian process for 500 iterations using the Adam optimiser with a learning rate of 0.01.

Appendix B

Additional Results

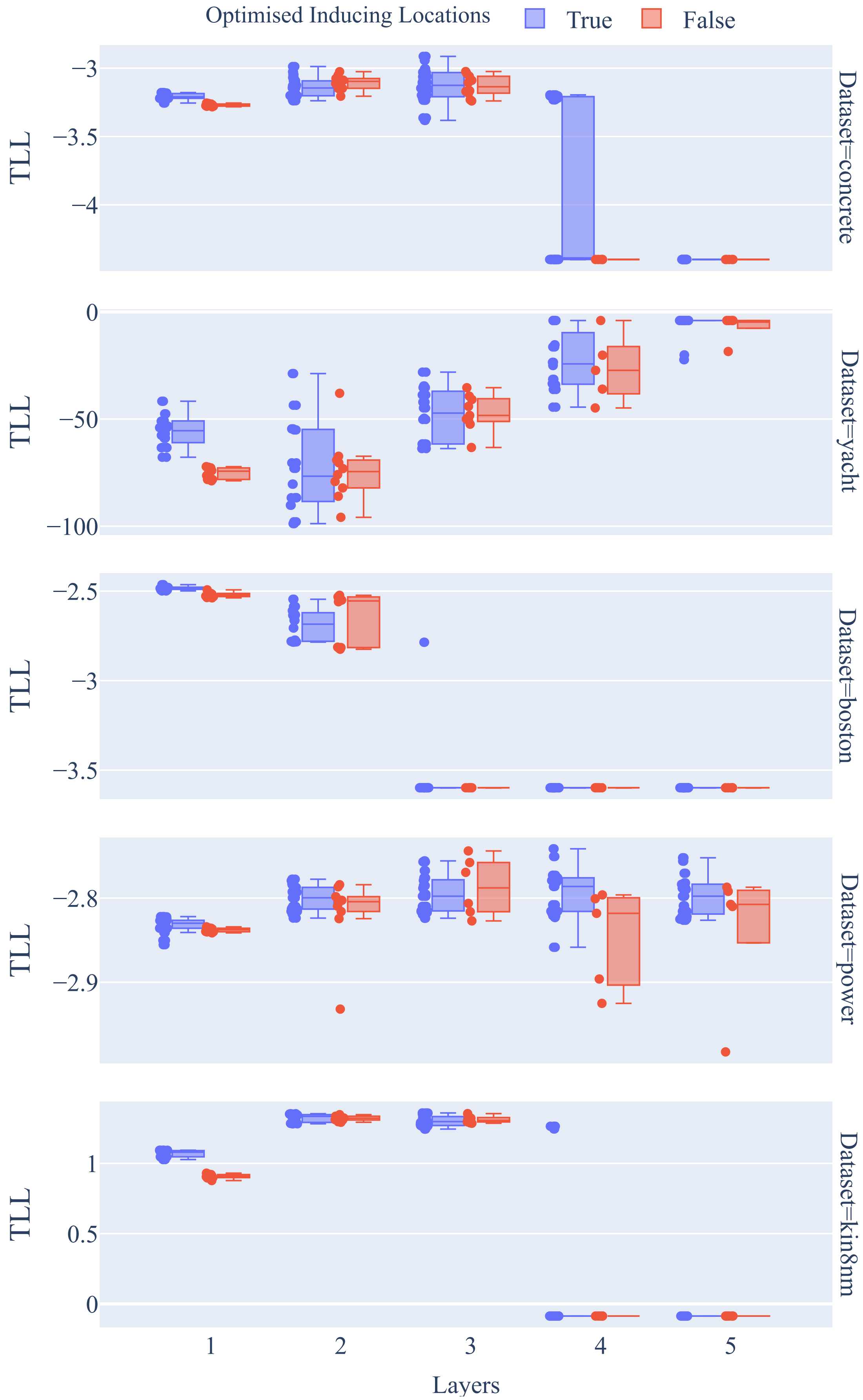

Figure B.1: Performance of Euclidean deep Gaussian processes on datasets from the UCI machine learning repository compared when inducing locations are and are not optimised during training. The number of layers in each model is given on the x-axis, with 1 indicating a shallow Gaussian process. Boxes show the interquartile range of the test-set performance over the 5 independent runs. The horizontal line inside each box indicates the median performance. Whiskers extend to the observations furthest away from the median but still within 1.5 times the interquartile range from either the lower