Imperial College London

BioMedIA

Inverse-free Inference and Reliable Uncertainty Quantification with Gaussian Processes

Author Sebastian Gabriel Popescu

Submitted 13th January 2022

Supervisors Prof. D. J. Sharp[1]
Dr. B. Glocker[2]
Dr. J. H. Cole[3]

[1]C3NL, Department of Medicine, Imperial College London, UK.
[2]BioMedIA, Department of Computing, Imperial College London, UK.
[3]CMIC, Department of Computer Science, University College London, UK.

A thesis submitted in fulfilment of the requirements for the degree of *Philosophiae Doctor*

Biomedical Image Analysis Group, Department of Computing,
Imperial College London

BioMedIA,
Department of Computing,
Huxley Building,
Imperial College London,
180 Queens Gate,
South Kensington,
SW7 2AZ

Sebastian G. Popescu © 2022

Never let the future disturb you. You will meet it, if you have to, with the same weapons of reason which today arm you against the present.

Marcus Aurelius Antoninus - *Meditations*

A stoic insight which often guided me through the highs and lows of this PhD.

To my parents, grandma and my cats.

Declaration

I, Sebastian Gabriel Popescu, hereby declare that this thesis is my own work unless otherwise specified. All published papers that are reproduced in this thesis are appropriately referenced and the reproduction adheres to the relevant copyright licenses. The thesis template has been adapted from Rob Robinson (Robinson 2020), who made his template available on his Github https://github.com/mlnotebook.

Copyright

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution 4.0 International Licence (CC BY).

Under this licence, you may copy and redistribute the material in any medium or format for both commercial and non-commercial purposes. You may also create and distribute modified versions of the work. This on the condition that you credit the author.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Prof. David J. Sharp, Dr. James H. Cole and Dr. Ben Glocker, for taking me as their PhD student, unwavering support through highs and lows and insightful guidance throughout the last five years. I would like to personally thank Dave for his patience and for believing in me during my time at C3NL, especially when working on more clinically focused papers that were way out of my comfort zone at the time. I would also like to thank James for his patience and understanding during the masters phase and first year of the PhD, when my work on *"BrainAge"* prediction was mediocre at best. I would also like to thank him for the flexibility shown in allowing me to pursue research topics which were completely outside the scope of the initial PhD proposal. Moreover, our weekly calls during the pandemic were of much help both to keep in touch with academic life in London but also due to the wide array of topics which we discussed. Lastly, I would like to thank Ben for introducing me to his group at BioMedIA, which had expanded my knowledge of machine learning at a crucial point in my PhD journey. Moreover, I am really grateful for your patience and understanding when at each meeting I would come with diametrically opposite research proposals during my "exploration" phase.

From C3NL I would like to thank Maria Balaet and Pedro da Costa for keeping me entertained both inside and outside the lab and a special thanks goes to Eyal Soreq, who has provided me with invaluable insights and help during my first months there. From BioMedIA I would like to give a shout-out to Kostas Kamnitsas and Nick Pawlowski, who helped me at key moments in my journey. Outside the labs, I would like to thank Steindór Sæmundsson and Hugh Salimbeni for insightful conversations on theoretical aspects of machine learning, Dan Iorga for much needed chit-chat in Romanian and Ahmad Khazaie for being my gym buddy. Lastly, a massive thanks to Mark van der Wilk, who in the last year has effectively acted as an extra supervisor. Without our almost weekly conversations, I am not sure that my last chapter would have been in its current shape.

Last but not least, I would like to thank my parents and my grandma for too many reasons to list here.

My PhD was funded by an EPSRC Centre for Doctoral Training studentship award to Imperial College London. The thesis template has been adapted from Rob Robinson (Robinson 2020), who made his template available on his Github https://github.com/mlnotebook.

Abstract

Standard machine learning research involves running experiments on training and testing data stemming from the same distribution, which is usually a data generating pipeline that involves clean data occurring in a clearly defined environment. However, in real-life scenarios models face unexpected distributions shifts, with the imperative need to discern unknown "unknowns", more specifically to detect outliers. Deep Neural Networks have proven their prowess in correctly classifying objects, albeit the lack of uncertainty, of a framework to incorporate prior knowledge and the reliance on large datasets. Bayesian methods are considered to fix said issues, with Gaussian Processes being an example of models that place function-space priors, finding usage due to their uncertainty quantification properties. Associated drawbacks to Gaussian Processes range from designing data-specific kernels to distributional mismatch between Gaussian predictive distribution and the real data generation distribution. This thesis posits that deep Gaussian Processes represent the optimal choice towards adequate out-of-distribution detection, as they circumvent aforementioned issues due to hierarchical nature. Firstly, we investigate deep Gaussian Processes' capabilities to properly detect outliers and propose changes to enhance out-of-distribution detection. Subsequently, we introduce a probabilistic layer that acts as a drop-in replacement for layers in convolutional architectures, with the property of reliably propagating uncertainty forward. We re-frame medical imaging prediction tasks as outlier detection, showing that our probabilistic module is more capable of detecting pathologies in MR scans as outliers given healthy samples in the training set. To ensure the competitiveness of our models we need to address the computational drawbacks associated to training Gaussian Processes. We propose an inverse-free variational lower bound to sparse Student-t Processes, showing through various experiments similar behaviour to matrix-inversion dependent models. Lastly, we dwell on future research pathways and applications, concluding that safe machine learning deployment is conditioned on probabilistic models with strong uncertainty guarantees.

Contents

1 Introduction **17**

1.1 Motivation: What uncertainties should we track? . . . 18
1.2 Motivation: Distance awareness and smoothness for out-of-distribution detection 26
1.3 Motivation: Hierarchical Gaussian Processes for out-of-distribution detection . . 29
1.4 Research Aims and Thesis Outline . . . 30

2 Background **33**

2.1 Gaussian Processes . . . 33
2.2 Sparse Approximations of Gaussian Processes . . . 37
2.2.1 Training Conditional approximations . . . 38
2.2.2 Variational Free Energy approximation . . . 39
2.2.3 Uncertainty decomposition in sparse GP through an evidential learning lens 43
2.3 Inverse-free Gaussian Processes . . . 46
2.3.1 Inverse-free Iterative Gaussian Processes . . . 46
2.3.2 Inverse-free Reparametrizable Sparse Variational Gaussian Processes . . 46
2.4 Deep Gaussian Processes . . . 48
2.4.1 Precursors to Deep Gaussian Processes . . . 48
2.4.2 Probabilistic definition of Deep Gaussian Processes . . . 48
2.4.3 Doubly Stochastic Variational Inference for Deep Gaussian Processes . . 51
2.4.4 Advances in approximate inference for Deep Gaussian Processes . . . 53
2.4.5 Pathologies in Deep Gaussian Processes . . . 53
2.5 Approximation of Gaussian Processes with noisy inputs . . . 58
2.5.1 Approximate Moments . . . 60
2.5.2 Exact Moments . . . 61

3 Distributional Deep Gaussian Processes **64**

3.1 Introduction . . . 64
3.2 Background . . . 65
3.2.1 Wasserstein-2 distance kernels on Gaussian measures . . . 66
3.2.2 Varifold theory or kernels between generalized measure spaces . . . 66
3.3 Propagation of variance in DGP . . . 67
3.4 Distributional Deep Gaussian Processes . . . 70

3.4.1 Desiderata for kernels in DGP . . . 70
3.4.2 Hybrid kernel . . . 71
3.4.3 Evidence lower bound . . . 74
3.4.4 Propagation of variance in DDGP . . . 76
3.4.5 *Feature-collapse* in (D)DGP . . . 76
3.5 Experiments . . . 78
3.6 Discussion . . . 84

4 Distributional Gaussian Processes Layers 87
4.1 Introduction . . . 88
4.2 Deep Wasserstein Kernel Learning . . . 91
4.2.1 Generative Process . . . 91
4.2.2 Evidence lower bound . . . 92
4.3 Distributional GP Layers . . . 93
4.3.1 Module architecture . . . 94
4.3.2 Imposing Lipschitz Conditions in Convolutionally Warped DistGP . . . 95
4.3.3 *Feature-collapse* in DistGP Layers . . . 97
4.3.4 Over-correlation in latent space . . . 98
4.3.5 Pooling operations on stochastic layers . . . 101
4.4 DistGP Layer Networks & OOD detection . . . 102
4.5 DistGP-based Segmentation Network & OOD Detection in Medical Imaging . . . 109
4.5.1 Data and pre-processing . . . 110
4.5.2 Brain tissue segmentation on normal MRI scans . . . 111
4.5.3 Outlier detection in MRI scans with tumors . . . 112
4.6 Discussion . . . 114

5 Matrix inversion-free inference in Student-t Processes 117
5.1 Introduction . . . 118
5.2 Background . . . 119
5.2.1 Distributions on positive definite matrices . . . 120
5.2.2 Student-t Processes . . . 121
5.2.3 Natural Gradients . . . 123
5.3 Generative process of conditional sparse Student's T Processes . . . 127
5.3.1 Generative process of sparse Student-t Processes . . . 127
5.3.2 Prior conditional matching in covariance matrices . . . 129
5.3.3 Variational Posteriors over Inverse-Wishart Processes . . . 131
5.4 Evidence lower bound of conditional sparse Student-t Processes . . . 131
5.5 Natural Gradients eliminate iterative bounds . . . 137

5.6 Experiments . . . 140
5.7 Discussion . . . 148
5.7.1 Relation to prior work . . . 148
5.7.2 Concluding remarks . . . 150

6 Conclusions **153**
6.1 Main Contributions . . . 154
6.2 Limitations & Future Research . . . 155
6.2.1 Uncertainty Decomposition & DGP variants . . . 155
6.2.2 Enhanced Approximate Inference for *really deep* DGP . . . 157
6.2.3 Refinements on inverse-free Student-t Processes . . . 157
6.2.4 Bayesian Optimization & Active Learning . . . 160
6.2.5 OOD detection & robustness to *"in-the-wild"* domain shifts . . . 161
6.2.6 Conclusion . . . 161

References **163**

Appendices **176**
A Further details pertaining to Distributional Deep Gaussian Processes . . . 176
A.1 Varifold Kernels . . . 176
A.2 Detailed proofs for Chapter 3 . . . 177
A.2.1 Positive-definite kernels using Wasserstein-2 distances . . . 177
A.2.2 Propagation of total uncertainty in DGP . . . 177
A.2.3 Propagation of distributional uncertainty in DGP . . . 179
A.2.4 Propagation of total uncertainty in DDGP . . . 180
A.2.5 Feature-collapse in DGP . . . 181
A.2.6 Feature-collapse in DDGP . . . 183
A.3 UCI experiments details . . . 184
A.4 Isotropic kernel properties in (Distributional) Deep Gaussian Processes . . . 185
A.5 Additional Results on Out-of-Distribution Detection for DDGP . . . 187
A.5.1 Sensitivity to input perturbation . . . 187
A.5.2 Additional Figures for OOD detection . . . 188
B Further details pertaining to Distributional GP Layers . . . 193
B.1 Detailed proofs for Chapter 4 . . . 193
B.1.1 Lipschitz properties of DistGP Layers . . . 193
B.1.2 Lipschitz properties of affine operator . . . 194
B.1.3 *Feature-collapse* in DistGP Layers . . . 195
B.2 Additional results on out-of-distribution detection . . . 198
B.2.1 Out-of-distribution detection in MRI scans with brain tumors . . . 198

B.2.2 Out-of-distribution detection in MRI scans with white matter hyperintensities . . . 198
B.2.3 Out-of-distribution detection on Retina scans . . . 199
C Further details pertaining to inverse-free sparse Student-t Processes . . . 206
C.1 Alternative Variational Posteriors over Inverse-Wishart Processes . . . 206
C.2 Variational lower bound of inverse free sparse Student-t Processes . . . 206
C.3 Obtaining predictions at testing time . . . 210
D Natural Gradient update rules for various distributions . . . 210
D.1 Natural Gradients for U . . . 211
D.2 Natural Gradients for T . . . 212

List of Figures

1.1 Conceptual difference between Epistemic and Aleatoric Uncertainty in Regression Tasks . . . 21
1.2 Conceptual difference between Epistemic and Aleatoric Uncertainty in Classification Tasks . . . 23
1.3 Decomposition of uncertainty in DNN-parametrized Dirichlet distributions . . . 25

2.1 GP prior samples. . . . 35
2.2 GP posterior samples. . . . 37
2.3 Probabilistic graphical models of DGP variants. . . . 52
2.4 Deep Gaussian Processes samples with zero mean function. . . . 55
2.5 Deep Gaussian Processes samples with identity mean function. . . . 56
2.6 Compositional uncertainty collapse in DSVI. . . . 57

3.1 Similarities between Varifold kernel and Hybrid kernel. . . . 67
3.2 Uncertainty in DGP is conditioned on first order derivatives. . . . 68
3.3 Conceptual differences between DGP and DDGP. . . . 72
3.4 Conceptual difference between euclidean and hybrid kernel. . . . 74
3.5 Non-parametric variance is correctly propagated in DDGP. . . . 78
3.6 Progressive collapse of distributional uncertainty in DGP . . . 78
3.7 Rationale behind collapse of non-parametric uncertainty. . . . 79
3.8 PCA mean functions influence distributional uncertainty in DGP. . . . 80
3.9 Smoothness comparison between DGP and DDGP . . . 81
3.10 Results on UCI datasets. . . . 82

4.1 Schematic of measure-preserving DistGP layer . . . 94
4.2 Influence of Lipschitz continuity . . . 103
4.3 Correlation in representation learning layer . . . 104
4.4 Reliability of *in-between* uncertainty . . . 105
4.5 Reliability of epistemic uncertainty . . . 105
4.6 Sensitivity to morphological deformations . . . 107
4.7 Sensitivity to image rotation . . . 108
4.8 Schematic of proposed DistGP activated segmentation net . . . 110
4.9 Schematic of study design. . . . 111
4.10 Case studies of brain tumor OOD detection . . . 113

4.11 Uncertainty on brain tumor tissue classification . . . 114

5.1 Convergence of inverse-free models on toy regression task . . . 140
5.2 Optimization trajectory of inducing locations on toy classification task . . . 142
5.3 Quality of approximate posterior of T. . . . 143
5.4 Inverse free models correctly recover optimal inducing points' locations . . . 144
5.5 Correct optimization of variational parameters . . . 145
5.6 Reliability of *in-between* uncertainty estimates . . . 145
5.7 Reliability of epistemic uncertainty estimates . . . 145
5.8 UCI results . . . 146
5.9 Effect of approximate posterior T on condition number . . . 147
5.10 Natural Gradient variant results . . . 147

A.1 Gaussian Annulus Theorem and behaviour of (D)DGP in high-dimensional spaces 186
A.2 Examples of OOD datasets . . . 187
A.3 Rotated-MNIST . . . 188
A.4 Morpho-MNIST . . . 189
A.5 OOD detection MNIST vs. NotMNIST . . . 190
A.6 OOD detection MNIST vs. SVHN . . . 191
A.7 OOD detection MNIST vs. FashionMNIST . . . 192
B.8 Detailed segmentation output for DistGP-Seg on BraTS . . . 199
B.9 Detailed segmentation output for OVA-DM on BraTS . . . 200
B.10 Intensity scale of WMH lesions. . . . 201
B.11 Detailed segmentation output for DistGP-Seg on WMH dataset . . . 202
B.12 Symptoms of Diabetic Retinopathy on retina scans. . . . 203
B.13 Detailed segmentation output for DistGP-Seg on DRIVE dataset . . . 204
B.14 Detailed segmentation output for DistGP-Seg on STARE/IDRID datasets . . . 205

List of Tables

3.1 Results on UCI datasets. 83
3.2 OOD detection results on MNIST using standard architectures. 84

4.1 Performance on MNIST and CIFAR-10 106
4.2 OOD detection results on MNIST using convolutional architectures. 107
4.3 Dice scores on tissue segmentation 111
4.4 Detecting brain tumors as outliers 112

Chapter 1

Introduction

A particular manner of defining artificial intelligence (AI) is as the study and design of rational agents, with a system being perceived as "intelligent" upon maximizing a certain notion of performance. The branch of AI that deals with tasks where experience is available to agents is machine learning. The empirical success of deep neural networks (DNN) in the 2010s brought upon us a new dawn in the pursuit to build AI. While using DNN, excellent performance has been achieved on well-defined narrow tasks with large amounts of data. However, these machine learning systems lack key qualities useful for safe and broad use in real-world applications, such as the communication of uncertainty in predictions, thereby essentially knowing when a system does not know what to do. One can rapidly see that such capabilities would be of paramount importance in applied settings such as healthcare or self-driving cars, where the outcome of the patient or the driver (alongside pedestrians and other drivers) hangs on the validity of predictions.

More often than not, performance is measured by how machine learning systems behave in novel situations that were not present within the data manifold of the training set. For example, in a biomedical setting, one trains a system to segment brain tumors on MRI scans stemming from either one or a finite set of scanning sites that are available. However, at testing time, it might face MRI scans from other scanning sites, which depending on how similar the respective scanner is with regards to the ones seen at training time, might result in a decrease of performance due to scanner effects. The capability of a machine learning system to generalize to new unseen data is measured by how small this gap in performance is.

With this in mind, there are a multitude of types of unknown/unseen/novel inputs that a machine learning system might encounter. Within the field of statistical learning the most commonly seen are in-distribution test points, the notion of in-distribution meaning that new data stems from the same data generating process as the training data, thereby making the two indistinguishable. Com-

ing back to our example with segmenting brain tumors, new MRI scans from unseen scanners represent in-distributions samples. An example of out-of-distribution data might be when training a machine learning system on a multi-site pooled datasets of MRI scans for the purpose of "brain-age" estimation, but at evaluation time the system is shown new subjects with brain tumors or white matter hyperintensities, which were not present at training time since we were training on normative data, more concretely subjects without any neurodegenerative diseases or pathologies. Devising machine learning algorithms that can detect these distribution shifts by virtue of their uncertainty estimates represents the primary focus of this thesis.

Now that we have laid out the main motivation of this thesis, we also need to specify which models are to be investigated. Gaussian Processes represent a Bayesian machine learning model that exhibit strong uncertainty quantification properties, making them a prime candidate to devise models which can detect distribution shifts. However, as we shall later see in the next chapter, Gaussian Processes suffer from some limitations. Hence, in this thesis we are concerned with building models from compositions of Gaussian Processes in a hierarchical manner. Studying their uncertainty quantification properties will represent a key goal of this thesis.

The remainder of this chapter is structured in a progressive manner, first motivating the different types of uncertainties and how to disentangle them. We then proceed to enunciate two key properties that any machine learning model needs to satisfy so as to be able to discern between in-distribution and out-of-distribution points. We then argue for the use of hierarchical compositions of Gaussian Processes to the detriment of DNN. Lastly, we present some key challenges associated to our Bayesian model of choice, which will constitute the basis of subsequent chapters in this thesis.

1.1 Motivation: What uncertainties should we track?

Before we delve into what uncertainty means, how to define it and how to disentangle it, we must first take a step back and consider how one acquires knowledge about objects and how valid that knowledge is.

Brief philosophical detour. Epistemology represents the philosophical study of the nature, origin, and limits of human knowledge. It attempts to study the following questions: what is the nature of knowledge?; what are the obstacles towards the attainment of knowledge?; what can be known?; how does knowledge differ from opinion or belief?. A key question pertaining to epistemology is how knowledge is acquired. The empirical method consists of acquiring it through experience or the senses (*e.g.*, one can examine a flower to infer certain characteristics, such as shape, smell, size, texture, whether it produced fruits or flowers). However, a natural follow-up question is whether we can trust our senses. It might also be that our sensory apparatus influences our inter-

pretation of reality. Epistemology proposes a dichotomy of types of knowledge: propositional or knowledge of facts (e.g., $2+2=4$) and practical knowledge (*e.g.*, knowing how to ride a bike). One might wonder at this point what has motivated such a pursuit. A simple answer is that one begins to question whether the ideas (*i.e.*, representations of the mind) correspond to the world outside of ourselves and if we can bridge this gap. Some philosophers have suggested that fear was the impetus of the development of epistemology: *"All epistemology begins in fear...fear that the senses are too feeble and the intellect too frail; fear that memory fades, even between adjacent steps of a mathematical demonstration; fear that authority and convention blinds; fear that God may keep secrets or demons deceive"* (Daston and Galison 1992). Lastly, a particularly extreme school of thought is skepticism, from the Greek word σκέπτομαι (skeptomai), which stands for to search, to think about or look for. This school of thought was divided into two: the academics, who believed that nothing could really be known about the world; and Pyrrhonism, who believed that the aforementioned academic position was knowledge in itself, thus trying and seeking to make the skeptical position logical in itself. A leading Phyrrhonist, Sextus Empiricus once affirmed *"Skepticism relieved two terrible diseases that afflicted mankind: anxiety and dogmatism."* (Empiricus 1933). In this thesis we will not attempt to relieve humanity of anxiety or dogmas, instead we will focus on devising machine learning systems that are innately skeptical about their predictions.

Perspectives on uncertainty. Now knowing what epistemology represents, we can find elements of it in representations of uncertainty in various fields of study. From a linguistic perspective, Ülkümen et al. (2016) split uncertainty into aleatoric and epistemic, with the former stemming from the Latin word for "dice", *alea*, and the latter latter type of uncertainty has its etymological roots in the Greek word πιστήμη (epistmē), which stands for knowledge. They argue that the distinction between aleatoric and epistemic uncertainty entails a difference in the way a statement is made, whether it is expressed in a confidence of likelihood language format. More specifically, speakers tend to use confidence statements to express epistemic uncertainty, such as "I am fairly confident" or "I am 90% sure". Conversely, people use likelihood statements to express aleatory uncertainty, such as "I believe it is fairly likely" or "I'd say there is a 90% chance". The authors consider a simple example by enunciating two simple questions. Firstly, if we consider a question of the type "Is the Amazon river longer than the Nile?", the authors make the case that uncertainty in this case stems from the gap in one's knowledge about a scientifically proven fact that is either true or false (*i.e.*, a case of propositional knowledge from epistemology). On the other hand, for a question of the type "Will a fair coin land heads in at least one of two flips?", the uncertainty stems from inherently stochastic behaviour of a physical device in the outside world.

From the realm of psychological analysis, more concretely of analysing responses to uncertainty from the viewpoint of expectation and surprise in perception, Kahneman and Tversky (1982) have also proposed this distinction between "inside" and "outside" uncertainty or between "internal" and "external" uncertainty as initially introduced in Howell and Burnett (1978), with the former being

attributed to the inner state of knowledge, whereas the latter is dependent on the dispositions of causal systems in the outside world. Kahneman and Tversky (1982) further distinguishes "external" uncertainty that is singular (an assessment of a specific case, *e.g.*, this team will win because it has great players) against distributional (an assessment of relative frequency of similar specific cases. *e.g.*, this player will succeed as he was trained in a prestigious sports academy).

The dichotomy of epistemic and aleatoric uncertainty was also proposed in Fox and Ülkümen (2011), where the authors characterize epistemic uncertainty as *"entailing missing information or expertise concerning an event that is, in principle, knowable."* The same authors characterize pure aleatory uncertainty as *"entailing as assessment of stochastic behavior that may be associated with a particular subjective probability but is otherwise unpredictable. Aleatory uncertainty is represented in relation to a class of possible outcomes, is focused on assessing an event's tendency to obtain, and is naturally measured by relative frequency."* The authors make the case that most events can be seen as a mixture of aleatoric and epistemic uncertainty. They bring up to discussion a simple example of the outcome of a football game, which can be seen as partly knowable given the relative strengths of teams, history of head to head matches and current team form, but it can also be considered as partly random as any outcome prediction needs to consider different conditions such as weather, mental state of players, arbitrary decisions by individuals. Finally, they stress out the subjective nature of the split between epistemic and aleatoric uncertainty.

For the purposes of this thesis, we shall consider uncertainty as defined in scientific modelling, by which predictions have a range of expected values, thus expressing the total uncertainty. We shall see that throughout the remainder of this thesis, we will only consider Gaussian distributions. This will inhibit us from reliably modelling extreme events (*e.g.*, stock market crash or extreme meteorological phenomena) and will be left as a limitation of subsequently introduced probabilistic models. Establishing the proportions in the range of expected values that are caused by aleatoric or epistemic uncertainty is the focus of the next paragraph.

Decomposition of uncertainty. In Der Kiureghian and Ditlevsen (2009), the authors make the case that problems involving analysis of reliability, risk assessment or decision-making are solved by defining a pool of possible models of how scientists and engineers envision natural occurring phenomena to be modelled using domain-specific knowledge. This pool of feasible models or distribution over models can contain intrinsic uncertainty, either caused by measurement errors or the imperfection of models to describe natural phenomena. The authors make the distinction between two types of uncertainties in a scientific context, aleatoric and epistemic. The former was first used in music composition to describe a succession of sounds that is pre-determined in general and follows a particular style, with details inserted that largely depend on chance (Meyer-Eppler 1958).

The authors identify the following sources of uncertainty:

- uncertainty in both input data and observations, caused by measurement errors (*i.e.*, aleatoric uncertainty)
- uncertainty caused by the selected form/architecture of the probabilistic model (*i.e.*, epistemic uncertainty)
- statistical uncertainty in the estimation of model parameters (*i.e.*, epistemic uncertainty)

From a machine learning perspective, Kendall and Gal (2017) propose a dichotomy of classifying sources of uncertainty into aleatoric or epistemic, depending on whether they can be reduced by adding more data or not. More specifically, aleatoric uncertainty is caused by inherent randomness in the data, hence it cannot be reduced (see Hüllermeier and Waegeman (2021) for an in-depth review of methods to quantify aleatoric and epistemic uncertainty in machine learning). The classical example of coin flipping is the most obvious one to give, even though there is research that supports that coin flipping is deterministic given a complex system taking into consideration strength of coin tosser, initial side, wind conditions and physics of coins tossing (Stefan and Cheche 2016; Strzałko et al. 2008). Conversely, epistemic uncertainty refers to the epistemic state of the agent/model, or more colloquially the ignorance with respect to the current state. This ignorance can be reduced by feeding the agent/model with more information/data.

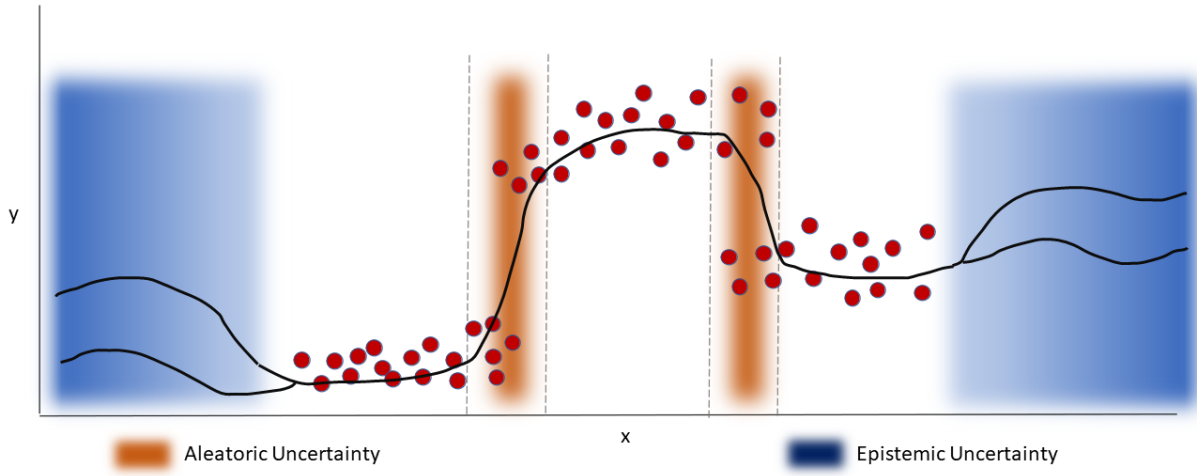


Figure 1.1: Conceptual difference between Epistemic and Aleatoric Uncertainty in Regression Tasks. Aleatoric uncertainty captures noise in the data generation process; Epistemic uncertainty captures uncertainty in model parameters, which can be caused by incomplete data in some input regions or due to domain shift; Intensity of colors denotes higher uncertainty in respective regions; With dark lines we denote the function-space samples of a hypothetical Bayesian machine learning algorithm fitted to the training data (red dots).

The goal is to decompose the total uncertainty into its constituent parts. One approach involves attributing the model's total predictive variance to the corresponding uncertainty sources using analytical methods. The first decomposition method relies on the analysis of variance (ANOVA), which is used to decompose a measure of total variation additively into its uncertainty sources (Bosshard et al. 2013; Garaud and Mallet 2011; Yip et al. 2011). A simpler and less convoluted

approach involves using a mixture of mutual information and expected entropy (Depeweg et al. 2018), which will be the default approach undertaken throughout the rest of this thesis.

Next we will have a closer look at the conceptual differences between aleatoric and epistemic uncertainties in simple regression or classification settings. In the first setting, considering input $x \in \mathbb{R}^D$ and a scalar output $y \in \mathbb{R}$, we assume an observation model with additive noise:

$$y = f(x) + \epsilon \, . \tag{1.1}$$

The total uncertainty, denoted by the variance $\mathbb{V}[y]$ contains both the residual error (*i.e.*, the variance of the observation error or aleatoric uncertainty) and the variance due to parameter uncertainty (*i.e.*, epistemic uncertainty).

One can easily decompose the variance into the following components:

$$\begin{aligned} \mathbb{V}[y] &= \mathbb{V}[f(x) + \epsilon] && (1.2) \\ &= \underbrace{\mathbb{V}[f(x)]}_{\text{Epistemic Uncertainty}} + \underbrace{\mathbb{V}[\epsilon]}_{\text{Aleatoric Uncertainty}} \, . && (1.3) \end{aligned}$$

One usually defines the additive noise as $\epsilon \sim \mathcal{N}(\epsilon \mid 0, \sigma^2_{noise})$. However, if one desires to model heteroskedastic noise, the additive noise can be made input-dependent $\epsilon \sim \mathcal{N}(\epsilon \mid 0, \sigma^2_{noise}(x))$.

$\mathbb{V}[\epsilon]$ is estimated as part of the model, by either learning σ^2_{noise} or $\sigma^2_{noise}(x)$. From (1.3) we also need to estimate $\mathbb{V}[f(x)]$, which is the estimated variance of the underlying function, which under current assumptions stands for epistemic uncertainty.

To more easily see in which scenarios these two sources of uncertainty can originate, we can take the toy example in Figure 1.1. Typically, a given model $f(x)$ will be trained using data covering a certain region of the input space, thus resulting in uncertainty outside of the training data manifold. In a regression setting, regions where there is higher variability in output space translate into higher aleatoric uncertainty.

Thus so far we have only approached uncertainty source decomposition from the point of view of regression. Equally important is the classification case. Hence, we consider a multi-class classification problem, with classes denoted as $\{y_1, \cdots, y_C\}$ and model parameters denoted by w, we have the following predictive equation at testing time:

$$p(y_c \mid x^*, \mathbf{D}) = \int p(y_c \mid x^*, w) p(w \mid \mathbf{D}) \, \mathrm{d}w \, . \tag{1.4}$$

Using the technique introduced in Depeweg et al. (2018), we proceed to decompose the two sources of uncertainty. The idea can be briefly described as quantifying the total uncertainty and aleatoric uncertainty, naturally the epistemic uncertainty following as the difference between the aforemen-

tioned. More specifically, they propose to measure the total uncertainty in terms of the entropy of the predictive posterior distribution, which in the case of discrete Y, is given by:

$$\mathcal{H}\left[y \mid x\right] = -\sum_{y \in Y} p(y \mid x) \log p(y \mid x) \; . \tag{1.5}$$

This measure of total uncertainty also contains the epistemic uncertainty about the network weights w. If we are to fix the set of weights (*i.e.*, considering a distribution $p\left(y \mid w, x\right)$), we essentially remove epistemic uncertainty. Depeweg et al. (2018) argues that by taking the expectation over the entropy of this varying distributions conditioned on different realisations of the weights w, we would arrive at a measure of aleatoric uncertainty, which we can now write as follows:

$$\mathbb{E}_{p(w|D)} \mathcal{H}\left[y \mid w, x\right] = -\int p(w \mid D) \left(\sum_{y \in Y} p(y \mid w, x) \log p(y \mid w, x) \right) \mathrm{d}w \; . \tag{1.6}$$

Epistemic uncertainty is obtained as the difference:

$$\mathcal{I}(y, w) = \mathcal{H}\left[y \mid x\right] - \mathbb{E}_{p(w|D)} \mathcal{H}\left[y \mid w, x\right] \; , \tag{1.7}$$

which represents the mutual information $\mathcal{I}(y, w)$ between y and w. Intuitively, epistemic uncertainty captures the amount of information pertaining to model parameters w that would be gained through knowledge of the true outcome y.

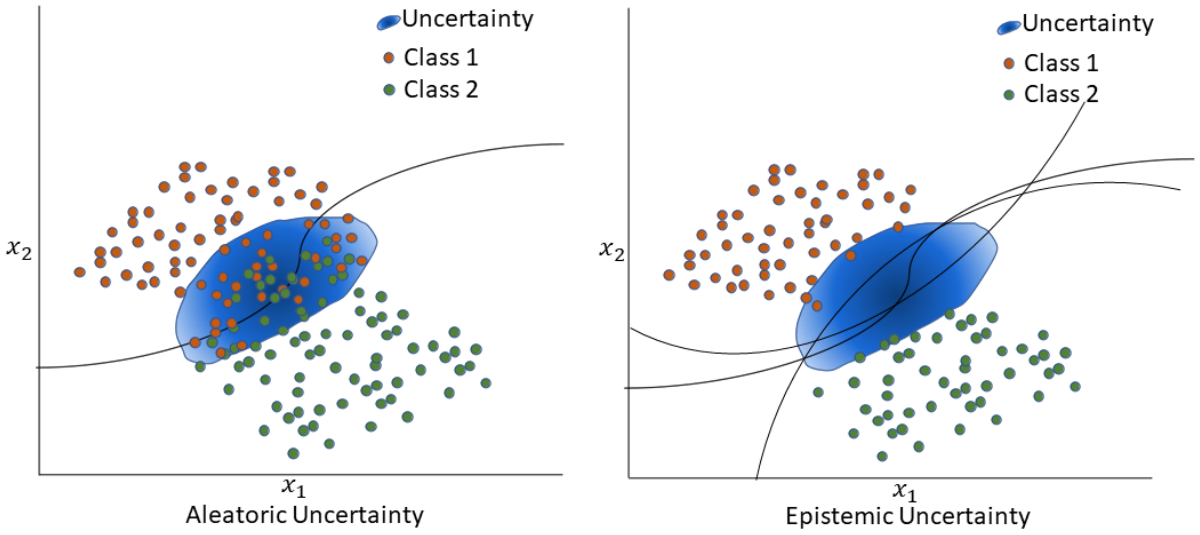


Figure 1.2: Conceptual difference between Epistemic and Aleatoric Uncertainty in Classification Tasks. Aleatoric uncertainty captures class overlap, which is irreducible; Epistemic uncertainty highlights regions in input space with a diverse hypothesis space of decision boundaries, which could be reduced given more data in respective input region; Shaded blue region denotes aleatoric (left) and epistemic uncertainty (right) given a different training set. With dark lines we denote the function-space samples of a hypothetical Bayesian machine learning algorithm fitted to the data.

To put these concepts to practical use, we can refer to a toy classification example as shown in Figure 1.2. Given a situation of class overlap, high aleatoric uncertainty will be assigned to said region. Conversely, in a different data regime, if the same region does not have class overlap but is on the path of the decision boundary between two classes, it might experience high variability in function samples due to all of these realisations being able to separate between the two classes.

A new type of uncertainty emerges. Generalization is of utmost importance for machine learning models, that is having trained models perform well on unseen data. In the case that the unseen data has different distributions (*i.e.*, a domain shift is present), this problem gets exacerbated (Quinonero-Candela et al. 2008; Torralba and Efros 2011), with even minuscule changes in the statistics of the testing set compared to the training set being able to lead CNN to fail completely (Kurakin et al. 2018). Therefore, properly detecting domain shift is key for the safe use of machine learning models.

Besides the dichotomy consisting of aleatoric and epistemic uncertainty, detecting certain inputs which have undergone a domain shift (Lakshminarayanan et al. 2017) or out-of-distribution (OOD) samples (Hendrycks and Gimpel 2016) has garnered a lot of interest in the past years. Succinctly, the aim is to measure the degree to which a model knows when it does not know. More precisely, the expectation for a domain shift aware model is that having been trained on a dataset with certain statistics, it will detect significant changes in data statistics at testing time by providing high predictive uncertainty on data points that are far away from the training data manifold.

A problem with introducing this new type of uncertainty is how to disentangle it from epistemic uncertainty. For example, in the Deep Ensembles paper (Lakshminarayanan et al. 2017), the authors propose to measure the disagreement between different sub-models of the deep ensemble $\sum_{m=1}^{M} \mathrm{KL}\left[p(y \mid x; \theta_m) \| \mathbb{E}\left[p(y \mid x)\right]\right]$ for M sub-models with associated sub-model parameters θ_m and $\mathbb{E}\left[p(y \mid x)\right] = \frac{1}{M} \sum_{m=1}^{M} p(y \mid x; \theta_m)$ is the prediction of the ensemble. We remind ourselves that epistemic uncertainty can be reduced by adding more data. By this logic, epistemic uncertainty cannot be reduced outside the data manifold of our dataset since we don't add data points which do not stem from the same data generative pipeline (this is not true in the case of OOD detection models which explicitly use OOD samples during training/testing (Hafner et al. 2020; Liang et al. 2017)). Hence, epistemic uncertainty can only be reduced inside the data manifold and should be zero outside the data manifold (under the assumption that the model is *distance-aware*, a concept subsequently introduced). Conversely, our chosen measure for OOD detection should grow outside the data manifold and be close to or 0 inside the data manifold. With this in mind, the disagreement metric introduced in Lakshminarayanan et al. (2017) cannot achieve this separation, confounding the two types of uncertainty.

Distributional Uncertainty. Malinin and Gales (2018) introduced for the first time the separation of total uncertainty into three components: epistemic, aleatoric and distributional uncertainty. To make the distinction clearer, the authors argue that aleatoric uncertainty is a "known-unknown", whereby the model confidently states that an input data point is hard to classify (*i.e.*, class overlap). Contrary, distributional uncertainty is an "unknown-unknown" due to the fact that the model is unfamiliar with the input space region that the test data comes from, thereby not being able to make confident predictions.

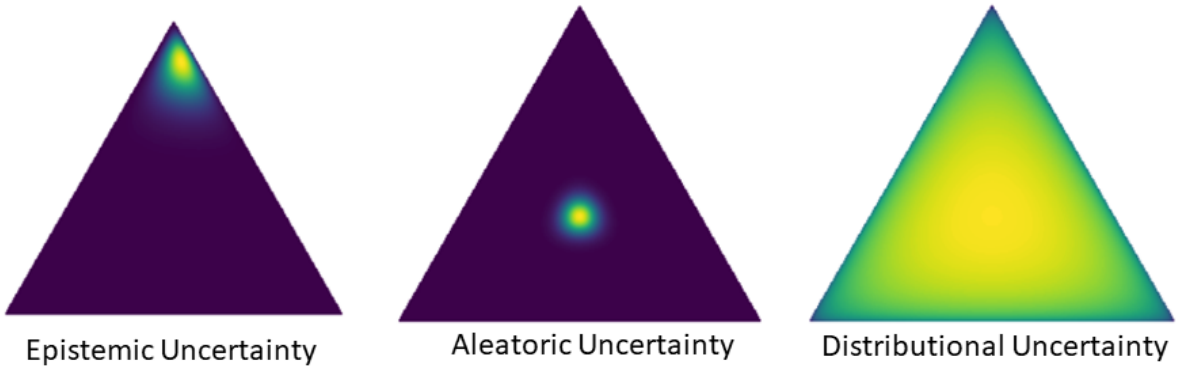


Figure 1.3: Decomposition of uncertainty in DNN-parametrized Dirichlet distributions. Probability simplex for a 3 class classification problem, where corners corresponds to a class; Each point represents a categorical distribution, with brighter colors indicating higher density of the underlying ensemble. Underlying machine learning model consists of a DNN that outputs the parameters of a Dirichlet distribution (see Malinin and Gales (2018)); **Epistemic Uncertainty** captures uncertainty in model parameters caused by lack of data or model non-identifiability, with the ensemble of the predictions being concentrated in a corner of probability simplex with an increased diversity of samples on the simplex corresponding to a higher epistemic uncertainty; **Aleatoric Uncertainty** captures class overlap, with the ensemble of predictions being confidently mapped to the highest predictive entropy, which is in the center of the simplex; **Distributional Uncertainty** captures domain shift, with the ensemble of predictions being centred in the middle of the simplex, with increasing degrees of diversity in samples (*i.e.*, estimated density is more spread out) resulting in higher distributional uncertainty;

We briefly introduce the uncertainty decomposition mechanism introduced in Malinin and Gales (2018). By using Monte Carlo integration in equation (1.4) and computing its predictive entropy (*i.e.*, $\mathcal{H}\left[y|x^*, D\right] = -\sum_{c=1}^{C} p(y_c|x^*, D) \log p(y_c|x^*, D)$), we would not be able to discern between high predictive entropy due to aleatoric uncertainty (*i.e.*, class overlap) or distributional uncertainty (*i.e.*, domain shift). Hence Malinin and Gales (2018) propose to introduce a latent variable μ over the categorical variables corresponding to each class, parametrized as a distribution over distributions on a simplex, $p(\mu \mid x^*, \theta)$, which for our purposes will be the Dirichlet distribution and where θ represent the parameters of the DNN that produces the parameters of the Dirichlet distribution (see Ulmer et al. (2023) for an extensive review of such models). The intuition behind this Dirichlet distribution over the probability simplex is that OOD points should be scattered, whereas in-distribution points should have samples on the simplex that concentrate. We can now re-write

our predictive equation as:

$$p(y_c|x^*, \mathbf{D}) = \int\int \underbrace{p(y_c|x^*, \mu)}_{\text{Aleatoric Uncertainty}} \underbrace{p(\mu|x^*, \theta)}_{\text{Distributional Uncertainty}} \underbrace{p(\theta|\mathbf{D})}_{\text{Epistemic Uncertainty}} \mathrm{d}\mu \, \mathrm{d}\theta \, . \tag{1.8}$$

The authors argue that using a measure of spread of the ensemble will be more informative, using the technique introduced in Depeweg et al. (2018). We remind ourselves that Mutual Information between variable X and Y can be expressed in terms of the difference between entropy and conditional entropy: $\mathcal{I}(X;Y) = \mathcal{H}(X) - \mathcal{H}(X|Y)$. Hence we can use the Mutual Information measure between model predictions and model parameters to obtain a better measure of uncertainty. We integrate out μ in the equation (1.8) and we obtain:

$$\underbrace{\mathcal{I}[y, \theta|x^*, \mathbf{D}]}_{\text{Epistemic Uncertainty}} = \underbrace{\mathcal{H}[\mathbb{E}_{p(\theta|\mathbf{D})} p(y|x^*, \theta)]}_{\text{Total Uncertainty}} - \underbrace{\mathbb{E}_{p(\theta|\mathbf{D})}[\mathcal{H}[P(y|x^*, \theta)]]}_{\text{Aleatoric Uncertainty}} \, . \tag{1.9}$$

The same mechanism is used to obtain distributional uncertainty. We integrate out over θ (in practice a single realisation of θ is considered) in equation (1.8) and we get:

$$\underbrace{\mathcal{I}[y, \mu|x^*, \mathbf{D}]}_{\text{Distributional Uncertainty}} = \underbrace{\mathcal{H}[\mathbb{E}_{p(\mu|\mathbf{D})} p(y|x^*, \mu)]}_{\text{Total Uncertainty}} - \underbrace{\mathbb{E}_{p(\mu|\mathbf{D})}[\mathcal{H}[P(y|x^*, \mu)]]}_{\text{Aleatoric Uncertainty}} \, . \tag{1.10}$$

Connections between this uncertainty disentanglement framework specialized for DNN-parametrized Dirichlet distributions and uncertainty disentanglement in GP will be subsequently made clearer in subsection 2.2.3. Distributional uncertainty will be the key uncertainty score used throughout this thesis to assess whether input data points are inside or outside the data manifold.

1.2 Motivation: Distance awareness and smoothness for out-of-distribution detection

Perhaps the greatest inspiration and motivation for this thesis resides in the theoretical framework introduced in Liu et al. (2020a), by which the authors outline what are some key mathematical conditions for provably reliable OOD detection in DNN. We commence by briefly outlining the ideas introduced in aforementioned paper.

For an abstract data generating distribution $p\left(y \mid x\right)$, where y is scalar, respectively $x \in \mathbb{X} \subset \mathbb{R}^D$ with the input data manifold being equipped with a suitable metric $\| \cdot \|_X$. We consider our training data $D = \{(x_i, y_i)_{1:n}\}$ to be sampled from a subset of the full input space $x_{in-d} \in \mathbb{X}$, where the *in-d* abbreviation stems from in-distribution. With this in mind, we can consider the in-distribution data generating distribution $p_{in-d}\left(y \mid x\right) = p\left(y \mid x, \, x \in \mathbb{X}_{in-d}\right)$, respectively the out-of-distribution data generating distribution $p_{ood}\left(y \mid x\right) = p\left(y \mid x, \, x \notin \mathbb{X}_{in-d}\right)$. Hence, it is

safe to assume the the full data generating distribution $p(y \mid x)$ is composed as a mixture of the in-distribution and OOD generating distributions:

$$p(y \mid x) = p(y, x \in \mathbb{X}_{in-d} \mid x) + p(y, x \notin \mathbb{X}_{in-d} \mid x) \tag{1.11}$$

$$= p(y \mid x, x \in \mathbb{X}_{in-d})\, p(x \in \mathbb{X}_{in-d}) + p(y \mid x, x \notin \mathbb{X}_{in-d})\, p(x \notin \mathbb{X}_{in-d}) \tag{1.12}$$

$$= p_{in-d}(y \mid x)\, p(x \in \mathbb{X}_{in-d}) + p_{ood}(y \mid x)\, p(x \notin \mathbb{X}_{in-d}) \ . \tag{1.13}$$

Evidently, during training we are only learning $p_{in-d}(y \mid x)$ since we only have access to $D \subset \mathbb{X}_{in-d}$. Therefore, our model is completely in the dark with regards to $p_{ood}(y \mid x)$. These two data generating distributions more often than not are fundamentally different (e.g., having trained a model on T1-weighted MRI scans, subsequently feeding it with T2-weighted MRI scans, an imaging modality which has an almost inverse scaling to represent varying brain tissue). With this in mind, Liu et al. (2020a) argue that the optimal strategy is for $p_{ood}(y \mid x)$ to be predicted as a uniform distribution, thus signalling the lack of knowledge of the model on this different input domain. We can now recall the distinction made by Malinin and Gales (2018), between "known-unknowns" (aleatoric uncertainty, *e.g.*, class overlap) and "unknown-unknowns" (distributional uncertainty, *e.g.*, domain shift), both of which have an uninformative predictive distribution (maximum predictive entropy). However, to disentangle these two types of uncertainty, we need a second-order type of uncertainty that basically scatters logit samples when distributional uncertainty is high, respectively accurately samples logits to maximum predictive entropy in the case of high aleatoric uncertainty (see Figure 1.3 for illustration). We now formalize this desideratum by a condition called "distance awareness" in Liu et al. (2020a).

Definition 1.2.1 (Definition 1 in Liu et al. (2020a))**.** We consider the predictive distribution for unseen point $p(y^* \mid x^*)$ at testing time, for model trained on $\mathbb{X}_{in-d} \in \mathbb{X}$, with the data manifold being equipped with metric $\|\cdot\|_X$. Then, we can affirm that $p(y^* \mid x^*)$ is *distance-aware* if there exists a summary statistic $u(x^*)$ of $p(y^* \mid x^*)$ that embeds the distance between $\mathbb{X}_{in-d}$ and x^*:

$$u(x^*) = v\left[\mathbb{E}_{x \sim \mathbb{X}_{in-d}}\left[\|x^* - x\|_X^2\right]\right] \ , \tag{1.14}$$

where v is a monotonic function that increases with distance.

GP predictive variance as distributional uncertainty. GP are clearly *distance-aware* provided we use a translation-invariant kernel. In this case we have $u(x^*) = K_{f^*f^*} - K_{f^*f} K_{ff}^{-1} K_{ff^*}$, which is monotonically increasing as a function of distance. Throughout this thesis, we will use the non-parametric variance of GP as a proxy for distributional uncertainty, which will be used to assess if inputs are in or outside the distribution.

Definition 1.2.1 does not make any assumptions related to the architecture of the model from which

the predictive distribution stems. In practice we would have the following composition to arrive at the logits $\text{logit}(x^*) = f \circ \text{enc}\,(x^*)$, where $\text{enc}\,(\cdot)$ represents a network that outputs the representation learning layer (*i.e.*, the penultimate layer in an architecture) and $f\,(\cdot)$ is the output layer. In Liu et al. (2020a) the authors propose the following two conditions to ensure that the composition is *distance-aware*:

- $f(\cdot)$ is *distance-aware*
- $\mathbb{E}_{x\sim\mathbb{X}_{in-d}}\left[\|x^* - X\|_X^2\right] \approx \mathbb{E}_{x\sim\mathbb{X}_{in-d}}\left[\|\,\text{enc}(x^*) - \text{enc}(X)\|_{\text{enc}(X)}^2\right]$

The last condition means that distances between data points in input space should be correlated with distances in the representation learning layer, which is equipped with a $\|\cdot\|_{\text{enc}(X)}$ metric. Whereas we have seen that for example a Gaussian Process satisfies the *distance-aware* condition for the last layer predictor, we are still left with the question on how to maintain distances in the representation learning layer correlated to distances in the input layer. This will be subsequently dealt with in Chapter 4.

Network smoothness constraints. Throughout this thesis we will consider the general term of "smoothness" of a model to mean the degree to which changes in the input have an effect on the output at a particular layer. The question now shifts into how can we quantify the smoothness of a network/function? In mathematical analysis a function $f : \mathbb{X} \to \mathbb{Y}$ is said to be k-smooth if the first k derivatives $\{f_\prime, f'', \cdots, f^{(k)}\}$ exist and are continuous. We denote functions which have these properties as being of class C^k. For example, Gaussian Processes using squared exponential kernels are C^∞ since the squared exponential kernel is infinitely differentiable. However, such a definition and quantification of *smoothness* would not aid us in ensuring the second condition of *distance-awareness*. For this, we shall use Lipschitz continuity, which is defined as follows: considering two metric spaces $\mathbb{X}$ and $\mathbb{Y}$ equipped with metrics $\|\cdot\|_X$ and $\|\cdot\|_Y$ and $f : \mathbb{X} \to \mathbb{Y}$ is Lipschitz continuous if $\exists$ real $K \geq 0$ such that $\forall x, y \in \mathbb{X}$ we have $\|f(x), f(y)\|_Y \leq K\|x, y\|_X$. Intuitively, for Lipschitz functions there is an upper limit in how much outputs can change with respect to distances in input space. It is perhaps better to highlight now that Lipschitz functions represent a global property. There are also locally Lipschitz continuous functions which respect the aforementioned condition just in a neighbourhood of x, respectively $B_r(x) = \{y \in \mathbb{X} : \|x, y\|_X \leq r\}$. Lastly, bi-Lipschitz continuity is defined as $\frac{1}{K}\|x, y\|_X \geq \|f(x), f(y)\|_Y \leq K\|x, y\|_X$, which is a property that avoids learning trivially smooth functions and maintains useful information (Rosca et al. 2020). With this in mind, recent work (Amersfoort et al. 2021; Liu et al. 2020a) have enforced the bi-Lipschitz property on feature extractors, thereby ensuring strong correlation between distances between data points in input space, respectively in the representation learning layer.

1.3 Motivation: Hierarchical Gaussian Processes for out-of-distribution detection

In the previous section we outlined the *distance-awareness* framework to ensure out-of-distribution capabilities for deep networks. We also briefly introduced recent *distance-aware* models that rely on bi-Lipschitz ResNet (He et al. 2016) feature extractors (*i.e.*, a network that lower and upper bounds the distortion of distances in the input space, see Liu et al. (2020a) for example), with a GP added as the last layer for performing predictions (Amersfoort et al. 2021). These models have proved to be efficient in detecting OOD samples, surpassing previous state-of-the-art methods such as Deep Ensembles (Lakshminarayanan et al. 2017). Much of the premise of this thesis resides in replacing deep feature extractors with fully probabilistic layers stacked on top of each other, with one such example being the deep Gaussian Processes (Damianou and Lawrence 2013). Throughout the remainder of this thesis, hierarchical GP and deep GP will be used interchangeably. The main motivation behind this line of inquiry is that GP are *distance-aware*, hence intuitively stacking GP on top of each other should in principle lead to a *distance-aware* feature extractor. However, this construction raises some questions and challenges, which we will now mention.

Do Hierarchical GP reliably propagate distributional uncertainty? Since their inception, research on deep GP has predominantly focused on enhancing their approximate inference framework. Little attention has been paid so far in how deep GP fare on OOD detection. Domingues et al. (2018) have proposed an autoencoder architecture with both encoder and decoder governed by deep GP to detect outliers. However, this line of inquiry stems from reconstruction-based methods for outlier detection (Baur et al. 2021), whereby at training time an autoenconder is trained on normative data, subsequently at testing time the model is presented with OOD samples. The differences between reconstruction and OOD sample represent the regions in input space which are anomalous. Conversely, in our line of research we are interested in accurate predictive models which also have OOD detection capabilities. Deep GP have thus so far not been tested if they are reliable OOD detection models. This will represent a major focus of this thesis.

How to integrate GP within a convolutional framework? GP for image classification have garnered interest in the past years, with hybrid approaches, whereby a deep neural network embedding mechanism is trained end-to-end with GP as the classification layer, being the first attempt to unify the two approaches (Bradshaw et al. 2017; Wilson et al. 2016). Garriga-Alonso et al. (2018) provided a conceptual framework by which classic CNN architectures are translated into the kernel of a shallow GP by exploiting the mathematical properties of the variance of weights matrices. Wilk et al. (2017) proposed the first convolutional kernel, constructed by aggregating patch response functions. Dutordoir et al. (2019) have attempted to solve the issue with complete spatial invariance of the convolutional kernel by adding an additional squared exponential kernel between the locations

of two patches to account for spatial location, obtaining improvements in accuracy. To extend this shallow GP model to accommodate deeper architectures, Blomqvist et al. (2018) have proposed to use the convolutional GP on top of a succession of feed-forward GP layers which process data in a convolutional manner akin to standard convolutional layers. However, scaling this framework to modern convolutional architectures with large number of channels in each hidden layer is problematic for two reasons. Firstly, this would imply training high-dimensional multi-output GP which still represents a research avenue on how to make it more efficient (Bruinsma et al. 2020). Ignoring correlations between channels would severely diminish the expressivity of the model. Secondly, the hidden layer GP which process data in a convolutional manner implies taking inducing points, with a dimensionality which scales linearly with the number of channels. This would imply optimization over high-dimensional spaces for each hidden layer, potentially leading to local minima. We will see later on (Chapter 4) an alternative to this framework for integrating GP in a convolutional architecture, one that is more amenable to modern convolutional architectures.

Matrix inversion-free inference in GP. Devising deep convolutional architectures using GP would imply computationally expensive matrix operations (*e.g.*, matrix inversion and log determinant) at each hidden layer. To enable hierarchical GP architectures for OOD detection to be competitive with neural network based counterparts we need to avert incurring these computational costs. Matrix inversion-free inference in GP has seen an increased interest in recent years (Gardner et al. 2018b; Wang et al. 2019; Wilk et al. 2020b). The last aim of this thesis will be to devise such inverse-free GP models and to ensure they have a predictive behaviour similar to matrix inversion-dependent counterparts.

1.4 Research Aims and Thesis Outline

Taking into account the aforementioned motivations, this thesis will deal with how to design computationally efficient *distance-aware* GP-based architectures that scale to modern convolutional architectures with the goal of devising models that provide both good predictive and OOD detection capabilities. We evaluate the OOD detection properties of our model on a set of biomedical image segmentation tasks.

1. Correctly propagating distributional uncertainty in Hierarchical Gaussian Processes Based on our assumption that the non-parametric variance of GP acts as a proxy for distributional uncertainty, we examine whether the non-parametric variance of GP gets propagated in a hierarchical architecture and what are the necessary conditions.

2. *Distance-aware* GP-based architectures in medical imaging tasks Can we scale hierarchical GP models to work on segmentation tasks within a biomedical imaging context? Can we leverage the distributional uncertainty so that clinicians can make better-informed decisions?

3. Matrix inversion-free inference in Student's T Processes Can we propose variational lower bounds to GP or Student-t Processes such that all operations are computationally efficient? Do we recover the same predictive behaviour at testing time as matrix inversion dependent counterparts?

This thesis will address the aforementioned research aims and is structured as follows.

Chapter 2 lays the foundations of the theoretical toolkit required for prospective readers to understand subsequent chapters assuming no prior knowledge on the subjects discussed. It introduces the fundamentals of Gaussian Processes and their sparse formulations which were designed to alleviate the computational burden of training said models, followed by an introduction to uncertainty disentanglement in sparse Gaussian Processes. Then we go on to to introduce recent attempts to devise matrix inversion-free inference frameworks for Gaussian Processes so as to decrease both training and testing times. Subsequently, we will introduce the most widely used approximate inference scheme for deep Gaussian Processes, a hierarchical multi-layered Gaussian Processes formulation, alongside some associated pathologies. Finally, we introduce a theoretical toolbox for analytical approximations of Gaussian Processes with uncertain inputs, which will be used to study the propagation of variance in deep Gaussian Processes.

Chapter 3 introduces a new hierarchical probabilistic model that is a variant of deep Gaussian Processes modified so that we operate in both Euclidean and Wasserstein space using a hybrid kernel. The rationale behind this kernel choice is that in the computation of Wasserstein distances, the geometric information present in the hidden layers is more carefully used, with benefits towards discerning outliers from inliers throughout the entire hierarchical architecture of deep Gaussian Processes. We demonstrate both theoretically and empirically that this reformulation of deep Gaussian Processes results in enhanced out-of-distribution detection capabilities for both tabular and image data.

- **Popescu, S. G.***, Sharp, D., Cole, J., and Glocker, B. (2020). "Hierarchical Gaussian Processes with Wasserstein-2 Kernels". In: *NeurIPS Workshop on Bayesian Deep Learning* – (Popescu et al. 2020)

Chapter 4 proposes a probabilistic module that can act as a drop-in replacement for the standard convolutional filters followed by an activation function present in standard machine learning models for image data. This module can be seen as an adaptation of the Distributional Gaussian

Process introduced in Chapter 3 to modern machine learning architectures. We derive Lipschitz constants for our module and show that by imposing an upper bound on the degree of change in output vis-à-vis to differences in input space, we can obtain a module that is *distance-aware*, thus satisfying one desideratum for efficient out-of-distribution detection. We construct segmentation networks with increased out-of-distribution detection abilities compared to state-of-the-art models from out-of-distribution detection literature on a range of biomedical imaging segmentation tasks.

- **Popescu, S. G.***, Sharp, D. J., Cole, J. H., Kamnitsas, K., and Glocker, B. (2021). "Distributional Gaussian Process Layers for Outlier Detection in Image Segmentation". In: *International Conference on Information Processing in Medical Imaging*. Springer, pp. 415–427 – (Popescu et al. 2021)

Chapter 5 introduces the generative process for sparse Student-t Processes. Based on this formulation, we augment the space of covariance matrices so as to obtain an easier to compute evidence lower bound that through construction is designed to be matrix inversion-free. We show that our proposed lower bound manages to retrieve the testing time predictive behaviour of matrix inversion dependent counterpart on a set of toy tasks, with promising results on real tabular datasets.

- **Popescu, S. G.***, Glocker, B., and Wilk, M. Van der (2022). "Matrix Inversion free variational inference in Conditional Student's T Processes". In: *Symposium on Advances in Approximate Bayesian Inference* – (Popescu et al. 2022)

Chapter 6 summarises the main contributions of this thesis. We continue by laying out limitations and open questions pertaining to our work, alongside some potential research avenues for the future.

Chapter 2

Background

This chapter provides a short introduction to the theoretical toolkit required for subsequent chapters. We commence by introducing what a Gaussian Process is, followed by a detailed chronological account of methods that circumvent the need for using the entire dataset for training and predictions, a process which is called the "sparsification" of Gaussian Processes. Subsequently, we introduce an uncertainty disentanglement framework for sparse Gaussian Processes, which will play a key role in future chapters in using these models for OOD detection. Whereas sparse GP already provide a significant speed up in terms of training time compared to full GP, further improvements in terms of time can be made by avoiding any matrix inversion operations altogether. With this in mind, we then briefly introduce matrix inversion-free inference frameworks for sparse Gaussian Processes.

We then explore variational inference training frameworks for deep Gaussian Processes and highlight some associated pathologies to such models, which will constitute a motivation for further developments in subsequent chapters. Lastly, we introduce a theoretical toolbox for analytical approximations of GP with noisy inputs, keeping in mind that the rationale is to better study the propagation of uncertainty in deep Gaussian Processes (which can be viewed as a succession of GP with noisy/uncertain inputs).

2.1 Gaussian Processes

A stochastic process assigns a random variable $f(x)$ to every element of an index set $\mathbb{X}$, more precisely $f = \{f(x)\}_{x\in\mathbb{X}}$. Stochastic processes can be defined via their finite dimensional marginal distributions, more concretely the stochastic process can be defined as a random function $f : \mathbb{X} \to \mathbb{Y}$, with any finite sequence $x_{1:n} = \{x_1, \cdots, x_n\}$ having a joint marginal distribution over

their values $y_{1:n} = \{y_1, \cdots, y_n\} = \{f(x_1), \cdots, f(x_n)\}$. Stochastic processes must respect two key conditions from the Kolmogorov Extension Theorem, which we now outline. Consider finite sequence $x_{1:n} \in \mathbb{X}$ with corresponding probability measure $p_{x_{1:n}}$ on $\left(\mathbb{R}^D\right)^n$.

Exchangeability. For all permutations π of $x_{1:n} = \{x_1, \cdots, x_n\}$ we impose $p_{x_{1:n}}(y_{1:n}) = p_{\pi(x_{1:n})}(\pi(y_{1:n}))$, which translates into invariance of the joint distribution with respect to permutations of elements in any given finite sequence.

Consistency. $p_{x_{1:m}}(y_{1:m}) = \int p_{x_{1:n}}(y_{1:n})\, \mathrm{d}y_{m+1:n}$, which ensures that if we marginalize a subset of elements, the remainder will remain unchanged.

Random functions are intrinsically complex mathematical objects with a Gaussian process representing the simplest random function due to its complete characterization by its mean and covariance functions (Pugachev 2013). A Gaussian process can be seen as a generalization of multivariate Gaussian random variables to infinite sets. We define this statement in more detail now. We consider $f(x)$ to be a stochastic field, with $x \in \mathbb{R}^D$ and we define $m(x) = \mathbb{E}[f(x)]$ and $C(x_i, x_j) = \mathrm{Cov}[f(x_i), f(x_j)]$. We denote a Gaussian process (GP) $f(x)$ as:

$$f(x) \sim \mathcal{GP}\left(m(x), C\left(x_i, x_j\right)\right) . \tag{2.1}$$

Throughout this thesis, we will generally consider zero-mean processes, hence our GPs are strictly characterized by their covariance functions. The latter have the condition to generate non-negative-definite covariance matrices, more specifically they have to satisfy:

$$\sum_{i,j} a_i a_j C\left(x_i, x_j\right) \geq 0 , \tag{2.2}$$

for any finite set $\{x_1, \cdots, x_n\}$ and any real valued coefficients $\{a_1, \cdots, a_n\}$.

Throughout this thesis we will only consider second-order stationary processes which have constant means and $\mathrm{Cov}[f(x_i), f(x_j)] = C(\|x_i - x_j\|)$. We can see that such covariance functions are invariant to translations.

The Matern form defines a general class of stationary covariance functions:

$$C^{Matern}(r) = \sigma^2 \frac{1}{\Gamma(\nu) 2^{\nu-1}} \left(\frac{2\sqrt{(}\nu) r}{l}\right)^{\nu} K_\nu \left(\frac{2\sqrt{\nu}}{l}\right) , \tag{2.3}$$

where $r = \|x_i - x_j\|$, Γ is the Gamma function, K_ν is the modified Bessel function of the second kind with $\nu > 0$ being the order of differentiability. The ν hyperparameter controls the level of smoothness of samples, which in this case are $\nu - 1$ differentiable. Another important kernel is

the Rational Quadratic, which has the following definition:

$$C^{RQ}(r) = \sigma^2 \left(1 + \frac{r^2}{2\alpha l^2}\right)^{-\alpha}, \tag{2.4}$$

which impose over sampled functions the property of varying smoothly across many lengthscales. The hyperparameter α indicates the relative weighting of large scale and small scale variations.

A special case of both Matern and Rational Quadratic covariance functions when $\nu \to \infty$ is the squared exponential/radial basis function kernel, which has the following form:

$$k^{SE}(x_i, x_j) = \sigma^2 \exp\left[\sum_{d=1}^{D} -\frac{(x_{i,d} - x_{j,d})^2}{l_d^2}\right], \tag{2.5}$$

where we have written its definition in the anisotropic case. Intuitively, the lengthscale values $\{l_1^2, \cdots, l_D^2\}$ represent the strength along a particular dimension of input space by which successive values are strongly correlated with correlation invariably decreasing as the distance between points increases. Such a covariance function has the property of Automatic Relevance Determination (ARD) (Neal 2012). However, the usage of ARD to determine the relative importance of a given feature in comparison to others has been recently challenged (Paananen et al. 2019). Lastly, the kernel variance σ^2 controls the variance of the process, more specifically the amplitude of function samples. In Figure 2.1 we provide examples of sampled GP functions from aforementioned kernels.

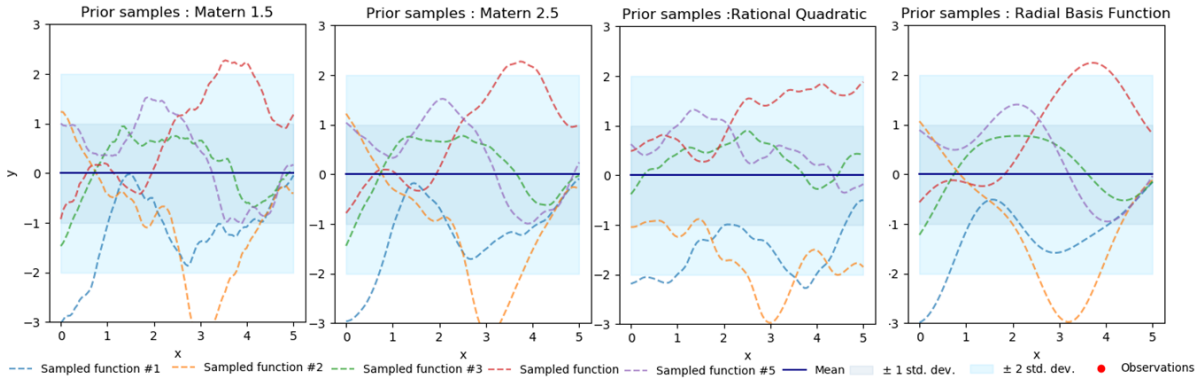


Figure 2.1: GP prior samples. Samples from GP priors with the Matern32, Matern52, RQ and RBF kernels. The shaded blue regions represent the standard deviations around the mean (dark blue line).

A GP has the following joint distribution over finite subsets $\mathbb{X}_1 \in \mathbb{X}$ with function values $f(X_1)$. Analogously for $\mathbb{X}_2$, with their union being denoted as $x = \{x_1, \cdots, x_n\}$.

$$p\begin{pmatrix} f(x_1) \\ f(x_2) \end{pmatrix} = \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ m(x_2) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1), k(x_1, x_2) \\ k(x_2, x_1), k(x_2, x_2) \end{bmatrix}\right). \tag{2.6}$$

The following observation model is used:

$$p(y|f,x) = \prod_{i=1}^{N} p(y_i \mid f(x_i)) \,, \tag{2.7}$$

where given a supervised learning scenario, the dataset $D = \{x_i, y_i\}_{i=1,\cdots,n}$ can be shorthand denoted as $D = \{x, y\}$. In the case of probabilistic regression, we make the assumption that the noise is additive, independent and Gaussian, such that the latent function $f(x)$ and the observed noisy outputs y are defined by the following equation:

$$y_i = f(x_i) + \epsilon_i \,, \tag{2.8}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2_{noise})$. To train a GP for regression tasks, one performs Marginal Likelihood Maximization of Type 2 over the following equation:

$$p(y) = \mathcal{N}\left(y \mid m, K + \sigma^2_{noise} I_n\right) \,, \tag{2.9}$$

by treating the kernel hyperparameters as point-mass. In the case of priors being placed on the hyperparameters, Markov Chain Monte Carlo (MCMC) methods can be used (Lalchand and Rasmussen 2020).

We are interested in finding the posterior $p\left(f(x^*) \mid y\right)$ since the goal is to predict for unseen data points x^* which are different than the training set. We know that the joint prior over training and testing set latent functions is given by:

$$p\begin{pmatrix} f(x) \\ f(x^*) \end{pmatrix} = \mathcal{N}\left(\begin{bmatrix} m(x) \\ m(x^*) \end{bmatrix}, \begin{bmatrix} k(x,x) & k(x,x^*) \\ k(x^*,x) & k(x^*,x^*) \end{bmatrix}\right) \tag{2.10}$$

and using marginalization properties of multivariate normals we can then write the following : $f(x^*) \sim \mathcal{N}\left(m(x^*), k(x^*,x^*)\right)$. With this in mind, and also reiterating that the marginal likelihood is also a multivariate normal given by $y \sim \mathcal{N}\left(m(x), k(x,x,) + \sigma^2_{noise} I_n\right)$, we can then write their joint prior since both of them are Gaussian:

$$p\begin{pmatrix} y \\ f(x^*) \end{pmatrix} = \mathcal{N}\left(\begin{bmatrix} m(x) \\ m(x^*) \end{bmatrix}, \begin{bmatrix} k(x,x) + \sigma^2_{noise} I_n & \mathrm{Cov}(y, f(x^*)) \\ \mathrm{Cov}(f(x^*), y) & k(x^*,x^*) \end{bmatrix}\right) \,. \tag{2.11}$$

We are now interested to see if we can simplify the covariance terms in the above equation:

$$\begin{aligned} \mathrm{Cov}\left[f(x^*), y\right] &= \mathbb{E}\left[\left[f(x^*) - m(x^*)\right]\left[y - m(x)\right]\right] & (2.12) \\ &= \mathbb{E}\left[\left[f(x^*) - m(x^*)\right]\left[f(x) + \epsilon - m(x)\right]\right] & (2.13) \end{aligned}$$

$$= \mathbb{E}\left[[f(x^*) - m(x^*)]\left[f(x) - m(x)\right]\right] + \mathbb{E}\left[\left(f(x^*) - m(x^*)\right)\epsilon\right] \tag{2.14}$$

$$= k(x^*, x^*) + 0 \tag{2.15}$$

Therefore, we can re-write our prior as:

$$p\begin{pmatrix} y \\ f(x^*) \end{pmatrix} = \mathcal{N}\left(\begin{bmatrix} m(x) \\ m(x^*) \end{bmatrix}, \begin{bmatrix} k(x,x) + \sigma^2_{noise} I_n & k(x,x^*) \\ k(x^*,x) & k(x^*,x^*) \end{bmatrix}\right). \tag{2.16}$$

Now we can simply apply the conditional rule for multivariate Gaussians to obtain:

$$p\left(f(x^*) \mid y\right) = \mathcal{N}(f(x^*) \mid m(x^*) + K_{f^*f}\left[K_{ff} + \sigma^2_{noise} I_n\right]^{-1}\left[y - m(x)\right], \tag{2.17}$$
$$K_{f^*f^*} - K_{f^*f}\left[K_{ff} + \sigma^2_{noise} I_n\right]^{-1} K_{ff^*}).$$

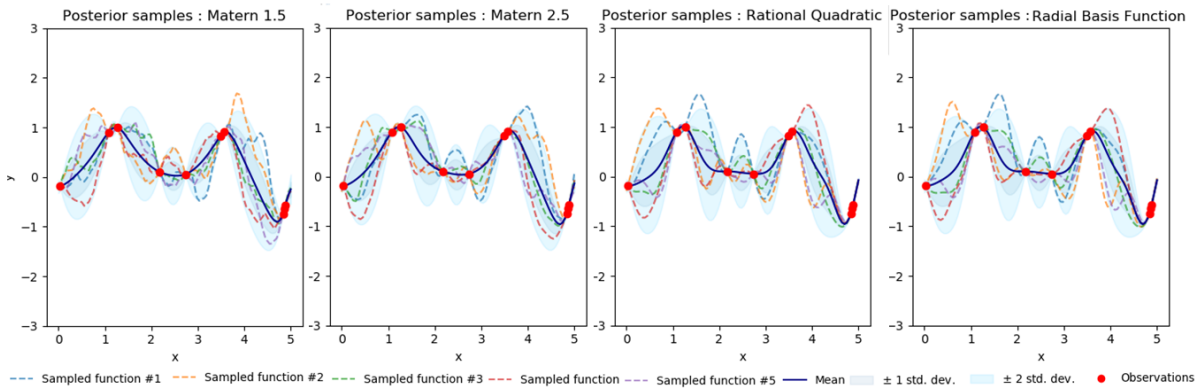


Figure 2.2: GP posterior samples. Samples from GP posteriors with the Matern32, Matern52, RQ and RBF kernels. The shaded blue regions represent the standard deviations around the mean (dark blue line).

Figure 2.2 shows examples of aforementioned predictive equation using an array of kernels. The usage of GP in real-life datasets is hindered by the $\mathcal{O}(N^3)$ time, $\mathcal{O}(N^2)$ memory for training, where N is the number of data points in the training set. In the next section we will see how to avert having to incur these expensive computational budgets.

2.2 Sparse Approximations of Gaussian Processes

As previously mentioned, using GP on large datasets is prohibitive due to the large computational cost attached to inverting covariance matrices. The fundamental idea behind this section is to construct artificial data points, that are not part of the training set, such that using just them we can closely recover the same training and testing time behaviour as if we used the entire dataset. One can essentially view these artificial data points as a compression of the full data.

2.2.1 Training Conditional approximations

In this subsection we succinctly review a historically commonly used probabilistic sparse approximation for Gaussian process regression. Quinonero-Candela and Rasmussen (2005) provides a unifying view of sparse approximations by placing each method into a common framework of analyzing their posterior and their *effective prior*, which will be shortly defined.

One way to avert the computationally expensive operators associated to the K_{ff} matrix is to modify the joint prior over $p(f, f^*)$ so that the respective terms depend on a matrix of lower rank. We introduce an additional set of M latent variables $U = \{U_1, \cdots, U_M\}$ with associated input locations $Z = \{Z_1, \cdots, Z_M\}$. Throughout this thesis, the former will be entitled inducing point values and the latter inducing point locations, respectively.

Due to the consistency of Gaussian processes, one can marginalize out U to recover the initial joint prior over $p(f, f^*)$:

$$p(f, f^*) = \int p(f, f^*, U)\, \mathrm{d}U = \int p(f, f^* \mid U) p(U)\, \mathrm{d}U\,, \tag{2.18}$$

where $p(U) = \mathcal{N}\left(U \mid 0,\ K_{uu}\right)$ and K_{uu} is the kernel covariance matrix evaluated at Z.

All subsequent sparse approximations introduced in this thesis originate from the following approximation:

$$p(f, f^*) \approx q(f, f^*) = \int q(f^* \mid U) q(f \mid U) p(U)\, \mathrm{d}U\,, \tag{2.19}$$

which translates into a conditional independency between training and testing latent variables given U. Intuitively, the name "inducing points" for $\{Z, U\}$ was given for this precise property, that U induces the values for the training and testing set.

For the purpose of comparison with more recent sparse approximations, we detail here the mathematical construction of Snelson and Ghahramani (2005), which proposed a method the authors called Sparse Gaussian Processes using Pseudo-inputs. However, we will refer to it using the naming conventions from Quinonero-Candela and Rasmussen (2005), hence throughout the thesis it shall be referred as the Fully Independent Training Conditional (FITC) approximation. This approximation method entails:

$$q_{FITC}\left(f|U\right) = \prod_{i=1}^{N} p\left(f_i|U\right) = \mathcal{N}\left(K_{fu}K_{uu}^{-1}U,\ \mathrm{diag}\left[K_{ff} - Q_{ff}\right]\right) \tag{2.20}$$

$$q_{FITC}\left(f_*|u\right) = p\left(F|U\right)\ . \tag{2.21}$$

These assumptions imply the following *effective prior*:

$$q_{FITC}\left(f, f^*\right) = \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}Q_{ff} - \operatorname{diag}\left[Q_{ff} - K_{ff}\right] & Q_{f,f^*}\\ Q_{f^*f} & K_{f^*f^*}\end{bmatrix}\right) . \tag{2.22}$$

We can notice that FITC imposes the exact covariances on the diagonal part of the covariance of the *effective prior*.

The predictive distribution follows naturally as:

$$q_{FITC}\left(f^*|y\right) = \mathcal{N}(Q_{f^*f}\left(Q_{ff} + \operatorname{diag}[K_{ff} - Q_{ff} + \sigma^2_{noise}I_n]\right)^{-1} y, \tag{2.23}$$

$$K_{f^*f^*} - Q_{f^*f}\left(Q_{ff} + \operatorname{diag}[K_{ff} - Q_{ff} + \sigma^2_{noise}I_n)\right)^{-1} Q_{ff^*} , \tag{2.24}$$

where we can notice the similarities with the predictive equation of GP (see equation (2.17)). The convergence to a full GP depends on the quality of the Nyström approximation (*i.e.*, Q_{ff}).

2.2.2 Variational Free Energy approximation

Titsias (2009) introduced the first variational lower bound comprising a probabilistic regression model over inducing points. More specifically, the authors applied variational inference (VI) in an augmented probability space that comprises training set latent function values F alongside inducing point latent function values U:

$$p(y, F, U) = p(y \mid F)p(F \mid U)p(U) , \tag{2.25}$$

where the conditional of the training set latent functions based on inducing points is taken to be exact $p(F \mid U) = \mathcal{N}\left(F \mid K_{fu}K_{uu}^{-1}U, K_{ff} - K_{fu}K_{uu}^{-1}K_{uf}\right)$. We would like to emphasize that this is in stark contrast to the approximation taken for FITC.

In terms of doing exact inference in this new model, respectively computing the posterior $p(f|y)$ and the marginal likelihood $p(y)$, it remains unchanged even with the augmentation of the probability space by U as we can marginalize $p(F) = \int p(F, U)\, \mathrm{d}U$ due to the marginalization properties of Gaussian processes. Succinctly, $p(F)$ is not changed by modifying the values of U, even though $p(F \mid U)$ and $p(U)$ do indeed change. This translates into the fundamental difference between variational parameters U and hyperparameters of the model $\{\sigma^2_{noise}, \sigma^2, l^2_1, \cdots, l^2_D\}$, whereby the introduction of more variational parameters does not change the fundamental definition of the model before probability space augmentation.

Collapsed Lower Bound. Our goal is to approximate the true posterior distribution $p(F, U \mid y) = p(F \mid U, Y)p(U \mid Y)$ by introducing a variational distribution $q(F, U)$ and minimizing

the Kullback-Leibler divergence:

$$\mathrm{KL}\left[q(F,U)\|p(F,U \mid y)\right] = \int q(F,U) \log \frac{q(F,U)}{p(F,U \mid y)} \,\mathrm{d}F \,\mathrm{d}U \;, \tag{2.26}$$

where the approximate posterior is factorized as $q(F,U) = p(F \mid U)q(U)$ and $q(U)$ is an unconstrained variational distribution over U. Following the standard VI framework we need to maximize the following variational lower bound on the log marginal likelihood:

$$\log p(y) \geq \int p(F|U)q(U) \log \frac{p(y|F)p(F|U)p(U)}{p(F \mid U)p(U)} \,\mathrm{d}F \,\mathrm{d}U \tag{2.27}$$

$$\geq \int q(U) \left[\int p(F|U) \log p(Y \mid F) \,\mathrm{d}F + \log \frac{p(U)}{q(U)}\right] \,\mathrm{d}U \;. \tag{2.28}$$

We can now solve for the integral over F:

$$\int p(F|U) \log p(y|F) \,\mathrm{d}F = \mathbb{E}_{p(F|U)} \left[-\frac{n}{2} \log(2\pi\sigma^2_{noise}) - \frac{1}{2\sigma^2_{noise}} \mathrm{Tr}\left[yy^\top - 2yF^\top + FF^\top\right]\right] \tag{2.29}$$

$$= -\frac{n}{2} \log(2\pi\sigma^2_{noise}) - \frac{1}{2\sigma^2_{noise}} \mathrm{Tr}[yy^\top - 2y\left(K_{fu}K_{uu}^{-1}U\right)^\top + \tag{2.30}$$

$$\left(K_{fu}K_{uu}^{-1}U\right)\left(K_{fu}K_{uu}^{-1}U\right)^\top + K_{ff} - Q_{ff}]$$

$$= \log \mathcal{N}\left(y|K_{fu}K_{uu}^{-1}U, \sigma^2_{noise}I_n\right) - \frac{1}{2\sigma^2_{noise}} \mathrm{Tr}\left[K_{ff} - Q_{ff}\right] \;. \tag{2.31}$$

We can now rewrite our variational lower bound as follows:

$$\log p(y) \geq \int q(U) \log \frac{\mathcal{N}\left(y \mid K_{fu}K_{uu}^{-1}U, \sigma^2_{noise}I_n\right) p(U)}{q(U)} \,\mathrm{d}U - \frac{1}{2\sigma^2_{noise}} \mathrm{Tr}\left[K_{ff} - Q_{ff}\right] \;. \tag{2.32}$$

One can differentiate this functional with respect to $q(U)$ so as to obtain an optimal value for this variational distribution (see equation (2.37)), subsequently reintroducing it in the variational bound.

Titsias (2009) argues for an equivalent approach of eliminating $q(U)$ from a variational lower bound, by "reversing Jensen's inequality" (see King and Lawrence (2006) for details about this trick). In brief, we have the following upper bound on $\int q(U) \log \frac{\mathcal{N}\left(y|K_{fu}K_{uu}^{-1}U, \sigma^2_{noise}I_n\right)p(U)}{q(U)} \,\mathrm{d}U = C$:

$$C \leq \log \int \frac{\cancel{q(U)}}{\cancel{q(U)}} \mathcal{N}\left(y \mid K_{fu}K_{uu}^{-1}U, \sigma_{noise}^2 I_n\right) p(U)\, \mathrm{d}U \tag{2.33}$$

$$\leq \log \int \mathcal{N}\left(y \mid K_{fu}K_{uu}^{-1}U, \sigma_{noise}^2 I_n\right) p(U)\, \mathrm{d}U\,. \tag{2.34}$$

Plugging this back into equation (2.32), we obtain:

$$\log p(y) \geq \log \int \mathcal{N}\left(y \mid K_{fu}K_{uu}^{-1}U, \sigma_{noise}^2 I_n\right) p(U)\, \mathrm{d}U - \frac{1}{2\sigma_{noise}^2} \operatorname{Tr}\left[K_{ff} - Q_{ff}\right]\,. \tag{2.35}$$

By using standard Gaussian linear transformation results the above becomes:

$$\log p(y) \geq \log \mathcal{N}\left(y|0, \sigma_{noise}^2 I_n + Q_{ff}\right) - \frac{1}{2\sigma_{noise}^2} \operatorname{Tr}\left[K_{ff} - Q_{ff}\right] = \mathcal{L}_{Titsias}\,. \tag{2.36}$$

Throughout this thesis, the above bound will be denoted as the Titsias bound or the collapsed lower bound for sparse Gaussian processes for regression (SGPR).

We can now derive the optimal $q^*(U)$ as follows:

$$q^*(U) \propto \mathcal{N}\left(y \mid K_{fu}K_{uu}^{-1}U, \sigma_{noise}^2 I_n\right) p(U)\,, \tag{2.37}$$

which after completing the quadratic form can be recognized as a multivariate Gaussian with the following formulation:

$$q^*(U) = \mathcal{N}(U \mid \sigma_{noise}^{-2} K_{uu}\left[K_{uu} + \sigma_{noise}^{-2} K_{uf}K_{fu}\right]^{-1} K_{uf}y, \\ K_{uu}\left[K_{uu} + \sigma_{noise}^{-2} K_{uf}K_{fu}\right]^{-1} K_{uu})\,. \tag{2.38}$$

Uncollaped Lower Bound. Previously we have derived a collapsed variational lower bound which involves using the entire training set to derive the lower bound to the log marginal likelihood, which is used for training the hyperparameters $\{\sigma_{noise}^2, \sigma^2, l_1^2, \cdots, l_D^2\}$ of the GP and for deriving the optimal posterior for U. Doing exact inference in the VFE formulation of sparse GP (SGP) is of sound theoretical interest, however it becomes infeasible for even medium sized datasets. In this subsection we will see how we can train VFE formulated SGP using minibatches by keeping an explicit representation over $q(U)$.

Stochastic Variational Inference (SVI) (Hoffman et al. 2013) enables the application of VI for extremely large datasets, by virtue of performing inference over a set of global variables, which induce a factorisation in the observations and latent variables, such as in the Bayesian formulation of Neural Networks with distributions (implicit or explicit) over matrix weights. GP do no exhibit

these properties, but by virtue of the approximate prior over testing and training latent functions for SGP approximations with inducing points U, which we remind here for the VFE scenario:

$$p(f, f^*) \approx q(f, f^*) = \int p(f|U)p(f^*|U)p(U)\,\mathrm{d}U\,. \tag{2.39}$$

This translates into a fully factorized model with respect to observations at training and testing time, conditioned on the global variables U. One can easily see that by integrating out U in the Titsias bound, this key property for SVI dissipates. In the remainder of this subsection, we will derive a lower bound to the Titsias bound that satisfies the SVI requirements, which was introduced in Hensman et al. (2013).

We reiterate the last equation from the derivation of the Titsias bound before integrating out U:

$$\log p(y) \geq \int q(U) \log \frac{\mathcal{N}\left(y \mid K_{fu}K_{uu}^{-1}U, \sigma^2_{noise}I_n\right) p(U)}{q(U)}\,\mathrm{d}U - \frac{1}{2\sigma^2_{noise}}\mathrm{Tr}\left[K_{ff} - Q_{ff}\right]\,. \tag{2.40}$$

The variational posterior is explicit in this case, respectively $q(F, U) = p(F|U; X, Z)q(U)$, where $q(U) = \mathcal{N}(U|m_U,\ S_U)$. Here, m_U and S_U are free variational parameters. Due to the Gaussian nature of both terms we can marginalize U to arrive at $q(F) = \int p(F|U)q(U)\,\mathrm{d}U = \mathcal{N}(F; \tilde{U}(x),\ \tilde{\Sigma}(x))$, where:

$$\tilde{U}(x) = K_{fu}K_{uu}^{-1}m_U \tag{2.41}$$

$$\tilde{\Sigma}(x) = K_{ff} - K_{fu}K_{uu}^{-1}\left[K_{uu} - S_U\right]K_{uu}^{-1}K_{uf}\,. \tag{2.42}$$

The lower bound can be re-expressed as follows:

$$\begin{aligned}\log p(y) &\geq \int q(U)\log\mathcal{N}\left(y|K_{fu}K_{uu}^{-1}U, \sigma^2 I_n\right)\,\mathrm{d}U - \mathrm{KL}\left[q(U)\|p(U)\right] &(2.43)\\ &- \frac{1}{2\sigma^2_{noise}}\mathrm{Tr}\left[K_{ff} - Q_{ff}\right]\\ &\geq \mathcal{N}\left(y|K_{fu}K_{uu}^{-1}m_U, \sigma^2_{noise}I_n\right) - \frac{1}{2\sigma^2_{noise}}\mathrm{Tr}\left[K_{fu}K_{uu}^{-1}S_UK_{uu}^{-1}K_{uf}\right] &(2.44)\\ &- \frac{1}{2\sigma^2_{noise}}\mathrm{Tr}\left[K_{ff} - Q_{ff}\right] - \mathrm{KL}\left[q(U)\|p(U)\right] = \mathcal{L}_{SVGP}\,,\end{aligned}$$

where we can easily see that the last equation factorized with respect to individual observations. This lower variational bound will be denoted as the Sparse Variational GP (SVGP) or uncollapsed SVGP bound and will constitute a stepping-stone in training the vast majority of new methods introduced by this thesis. Nevertheless, SVGP still require costly computational operations in its bound (*e.g.*, matrix inversion and log determinant of K_{uu}). Later in this section we will see methods which attempt to avert computing these exact matrix operations to as to speed up training time.

Behaviour of FITC vs. SVGP In Bauer et al. (2016) the authors showcase differences in the predictive behaviour of FITC and SVGP on an array of toy regression tasks. We briefly summarize their findings and contextualize them within the framework of uncertainty decomposition introduced in section 1.1. In extreme scenarios, FITC will incorrectly estimate σ^2_{noise} to be almost zero, hence FITC might exhibit pathological behaviour in estimating aleatoric uncertainty in regression tasks. Conversely, SVGP tends to overestimate σ^2_{noise}, making little use of input-dependent uncertainty (epistemic uncertainty). FITC can place inducing points' locations on top of each other so that it averts losing the benefits of non-input dependent uncertainty (aleatoric uncertainty). Putting this observation in the context of estimating distributional uncertainty, adding inducing points in FITC does not necessarily result in a better separation between in-distribution points and OOD points. Lastly, SVGP's posterior predictive distribution and marginal likelihood approximations become more accurate upon adding inducing points, with FITC not generally recovering the full GP. For all these reasons, throughout the remainder of this thesis we will use SVGP as the default sparse approximation of a full GP.

2.2.3 Uncertainty decomposition in sparse GP through an evidential learning lens

In subsection 1.1 we have introduced the rationale behind the uncertainty decomposition framework introduced in Malinin and Gales (2018). We now expand on this topic on how to separate uncertainties in deep evidential learning models (Amini et al. 2019; Sensoy et al. 2018) and make an analogy to how uncertainties are decomposed in SVGP.

For multi-class classification tasks, evidential learning directly parametrizes predictive distributions over the probability simplex (see Ulmer et al. (2023) for an introduction to deep evidential learning). Hence, in comparison to Bayesian deep learning (BDL) or Deep Ensembles (Lakshminarayanan et al. 2017) it averts parametrizing the logit space, subsequently feeding it through a softmax function. Dirichlet distributions provide an obvious choice for defining a distribution over the $K-1$ dimensional probability simplex, having the following p.d.f.: $\mathrm{Dir}\,(x \mid \alpha) = \frac{1}{\beta(\alpha)} \prod_{c=1}^{K} x_c^{\alpha_c - 1}$, where $\beta(\alpha) = \frac{\prod_{c=1}^{K} \Gamma(\alpha_c)}{\Gamma(\alpha_0)}$ and $\alpha_0 = \sum_{c=1}^{K} \alpha_c$ with $\alpha_c \geq 0$. α_0 is called the precision, being similar to the precision of a Gaussian distribution, where larger α_0 will indicate a sharper distribution.

Dirichlet networks involve having a NN predict the concentration parameters of the Dirichlet distribution $\alpha = f_\theta(x)$, where predictions are made as follows: $\tilde{y} = \arg\max_c \{\frac{\alpha_c}{\alpha_0}\}_{c=1}^{K}$.

We remind ourselves the uncertainty decomposition framework laid out in subsection 1.1:

$$\underbrace{\mathcal{I}[y, \mu | x^*, \mathbf{D}]}_{\text{Distributional Uncertainty}} = \underbrace{\mathcal{H}[\mathbb{E}_{p(\mu|\mathbf{D})} p(y|x^*, \mu)]}_{\text{Total Uncertainty}} - \underbrace{\mathbb{E}_{p(\mu|\mathbf{D})}[\mathcal{H}[P(y|x^*, \mu)]]}_{\text{Aleatoric Uncertainty}} . \tag{2.45}$$

In the case of Dirichlet networks these uncertainty measures have analytic formulas (Malinin and Gales 2018):

$$\mathbb{E}_{p(\mu|\mathbf{D})}[\mathcal{H}[P(y|x^*,\mu)]] = -\sum_{c=1}^{K} \frac{\alpha_c}{\alpha_0} \left[\psi(\alpha_c+1) - \psi(\alpha_0+1)\right] \tag{2.46}$$

$$\mathcal{I}[y,\mu|x^*,\mathbf{D}] = -\sum_{c=1}^{K} \frac{\alpha_c}{\alpha_0} \left[\log \frac{\alpha_c}{\alpha_0} - \psi(\alpha_c+1) + \psi(\alpha_0+1)\right], \tag{2.47}$$

where ψ is the digamma function. Epistemic uncertainty quantifies the spread in the Dirichlet distribution, hence α_0 (Charpentier et al. 2021) can be used to measure it since the precision is a measure of how sharp the distribution is.

Exact inference is not tractable in GP on classification tasks due to the non-conjugacy between the GP prior and the non-Gaussian likelihood (Categorical or Bernoulli). Therefore, approximations are required such as the Laplace approximation (Williams and Barber 1998), Expectation Propagation (Hernández-Lobato and Hernández-Lobato 2016; Minka 2013) or Variational Inference (Hensman et al. 2015). Milios et al. (2018) have proposed a method that circumvents these approximate inference techniques by rebranding the classification problem into a regression one, for which exact inference is possible. We commence to briefly lay out the Dirichlet-based GP Classification algorithm.

We consider the probability simplex $\pi = \{\pi_1, \cdots, \pi_K\} \sim \mathrm{Dir}(\alpha)$. We can transform a multi-class classification task into an independent multi regression scenario. If we consider a one-hot-encoding of the form $y_c = 1$, then we can assign $\alpha_c = 1 + \alpha_\epsilon$, respectively $\alpha_k = \alpha_\epsilon$ for $\forall k \neq c$ with $0 \leq \alpha_\epsilon \ll 1$. The model has the following generative process:

$$\pi \sim \mathrm{Dir}(\alpha) \tag{2.48}$$

$$p(y \mid \alpha) = \mathrm{Cat}(\pi) \tag{2.49}$$

To sample from the Dirichlet distribution we use the following routine: $\pi_c = \frac{x_c}{\sum_{k=1}^{K} x_k}$ with $x_c \sim \Gamma(\alpha_c, 1)$ following the Gamma distribution. From this sampling procedure, we can see that the generative process translates to independent Gamma likelihoods for each class. Intuitively, at this point in the derivation we need GPs to produce $\{x_c \geq 0\}_{c=1}^{K}$, since Gamma distributions are only defined on $\mathbb{R}^+$. Since the marginal GP over a finite subset of data is governed by a multivariate normal it will not satisfy this constraint. To obtain sampled GP functions that respect this constraints, we can use an $\exp$ function to transform it. With this in mind, we know that $x \sim \text{log-normal}\,(x \mid \mu, \sigma^2) \overset{d}{=} \exp y$ for $y \sim \mathcal{N}(y \mid \mu, \sigma^2)$, with $\mathbb{E}[x] = \exp\left(\mu + \frac{\sigma^2}{2}\right)$ and $\mathbb{V}[x] = [\exp\sigma^2 - 1]\exp(2\mu + \sigma^2)$. Hence, we can approximate $x_c \sim \Gamma(\alpha_c, 1)$ with $\tilde{x_c} \sim \text{log-normal}\left(\tilde{\mu}_c, \tilde{\sigma}_c^2\right)$. To ensure a good approximation, the authors in Milios et al. (2018) propose

using moment matching:

$$\mathbb{E}\left[x_c\right] = \alpha_c = \exp\left(\tilde{\mu}_c, \frac{\tilde{\sigma}_c^2}{2}\right) = \mathbb{E}\left[\tilde{x}_c\right] \tag{2.50}$$

$$\mathbb{V}\left[x_c\right] = \alpha_c = \left[\exp\tilde{\sigma}_c^2 - 1\right]\exp\left[2\tilde{\mu}_c + \tilde{\sigma}_c^2\right] = \mathbb{V}\left[\tilde{x_c}\right] \ , \tag{2.51}$$

with equality if $\tilde{\mu}_c = \log\alpha_c - \frac{\tilde{\sigma}_c^2}{2}$ and $\tilde{\sigma}_c^2 = \log\left(\frac{1}{\alpha_c} + 1\right)$. We can re-express this approximation by taking a natural logarithm, obtaining $\log\tilde{x}_c \sim \mathcal{N}\left(\tilde{\mu}_c, \tilde{\sigma}_c^2\right)$. This translates into a heteroskedastic regression model $\tilde{\mu}_c = f_c + \mathcal{N}\left(0, \tilde{\sigma}_c^2\right)$, where $f_c \sim \mathcal{GP}\left(0, K_{ff}\right)$. Hence, one can now apply the standard inference scheme for full GP or we can sparsify the model and apply the SVGP framework. At testing time, the expectation of class probabilities will be:

$$\mathbb{E}\left[\pi_{i,c}\right] = \int \frac{\exp f_{i,c}}{\sum_{k=1}^{C}\exp f_{i,k}} q(f_{i,c})\,\mathrm{d}f_{i,c}\ , \tag{2.52}$$

which can be approximated via Monte Carlo integration. In the sparse scenario, we have $q(f_{i,c}) \sim \mathcal{N}\left(\tilde{U}(x_i), \tilde{\Sigma}(x_i)\right)$ similar to the predictive equations (2.41) and (2.42). Therefore, if we use Dirichlet-based GP for classification tasks we can obtain similar estimates of aleatoric and distributional uncertainty in the space of the probability simplex as in equations (2.46) and (2.47) specific to Dirichlet Networks. However, for the purposes of this thesis we intend to measure distributional uncertainty in the space of logits, as the formulas are simpler to compute and more intuitive from the viewpoint of *distance-awareness*.

As we have previously stated, GPs are *distance-aware*. Thus, they can reliably notice departures from the training set manifold. For SVGP we decompose the model into two components:

$$h(x) \sim \mathcal{N}(h|0, K_{ff} - K_{fu}K_{uu}^{-1}K_{uf}) \tag{2.53}$$

$$g(x) \sim \mathcal{N}(g|K_{fu}K_{uu}^{-1}m_U, K_{fu}K_{uu}^{-1}S_UK_{uu}^{-1}K_{uf}) \tag{2.54}$$

The $h(x)$ variance captures the shift from within to outside the data manifold and will be denoted as *distributional uncertainty* (interchangeably also denoted as non-parametric variance). The variance $g(\cdot)$ is denoted as *epistemic uncertainty* (interchangeably also denoted as parametric variance) ,encapsulating uncertainty present inside the data manifold. To capture the overall uncertainty in $h(x)$, thereby also capturing the spread of samples from it, we can calculate its differential entropy as:

$$\mathcal{H}\left[h(x)\right] = \frac{n}{2}\log 2\pi + \frac{1}{2}\log\left|K_{ff} - K_{fu}K_{uu}^{-1}K_{uf}\right| + \frac{1}{2}n\ . \tag{2.55}$$

In practice we only use the diagonal terms of the Schur complement, hence the log determinant term will considerably simplify. Intuitively, if terms on the diagonal of the Schur complement have higher values, so will the distributional differential entropy. This OOD measure in logit space will be

used throughout the rest of the thesis for measuring departures from training set manifold.

2.3 Inverse-free Gaussian Processes

In the previous section we have explored sparse GP approximations, which have reduced the computation cost to $\mathcal{O}\left(M^3\right)$. Now, we briefly introduce research done on going on step further, meaning in creating sparse GP that do not even require inverting the covariance matrix related to inducing points Z, thus considerably reducing the computational cost associated to training such models.

2.3.1 Inverse-free Iterative Gaussian Processes

We remind ourselves that for a GP the marginal likelihood can be expressed as follows:

$$\log p\left(y \mid X, \theta\right) \propto -y^{\top}\left(K_{ff}+\sigma_{noise}^{2} I_{n}\right)^{-1} y-\log\left|K_{ff}+\sigma_{noise}^{2} I_{n}\right| . \tag{2.56}$$

Optimizing it with respect to kernel hyperparameters θ, we arrive at:

$$\begin{aligned}\frac{\partial \log p\left(y \mid X, \theta\right)}{\partial \theta} \propto y^{\top}\left(K_{ff}+\sigma_{noise}^{2} I_{n}\right)^{-1} \frac{\partial K_{ff}+\sigma_{noise}^{2} I_{n}}{\partial \theta}\left(K_{ff}+\sigma_{noise}^{2} I_{n}\right)^{-1} y- \\ \operatorname{Tr}\left[\left(K_{ff}+\sigma_{noise}^{2} I_{n}\right)^{-1} \frac{\partial K_{ff}+\sigma_{noise}^{2} I_{n}}{\partial \theta}\right] .\end{aligned} \tag{2.57}$$

The following terms $y^{\top}\left(K_{ff}+\sigma_{noise}^{2} I_{n}\right)^{-1}$ and the trace term are computationally expensive operations that are solved through the use of Conjugate Gradients (CG) (see Davies (2015) for an in-depth introduction) such as in recent work (Gardner et al. 2018b; Wang et al. 2019). These methods were shown to work quite efficiently for full GP. Throughout this thesis, this line of work will be collectively gathered under the umbrella term of Iterative Gaussian Processes since evaluating the log marginal likelihood involves an iterative procedure specific to CG to arrive at a solution close to the real inverse $\left(K_{ff}+\sigma_{noise}^{2} I_{n}\right)^{-1}$.

2.3.2 Inverse-free Reparametrizable Sparse Variational Gaussian Processes

The goal of inverse-free methods is cut the computational cost $\mathcal{O}\left(BM^2+M^3\right)$ associated to SVGP to one which is approximately $\mathcal{O}\left(M^2\right)$, where B represents the size of the mini-batch. We reiterate the uncollapsed SVGP bound:

$$\mathcal{L}_{SVGP}=\sum_{i=1}^{N} \mathbb{E}_{q(f(x_i))}\left[\log p(y_n \mid f(x_n))\right]-\mathrm{KL}\left[q(U) \| p(U)\right] , \tag{2.58}$$

where $q(f_{x_i}) = \mathcal{N}\left(f_i \mid \tilde{U}(x_i), \tilde{\Sigma}(x_i)\right)$, with $\tilde{U}(x_i) = K_{f_iu}K_{uu}^{-1}m_U$ and $\tilde{\Sigma}(x_i) = K_{f_if_i} - K_{f_iu}K_{uu}^{-1}K_{uf_i} + K_{f_iu}K_{uu}^{-1}S_UK_{uu}^{-1}K_{uf_i}$. The goal of inverse-free methods is to eliminate the need to compute K_{uu}^{-1} in these equations.

In Wilk et al. (2020b), the authors propose the following reparametrization to eliminate matrix inverses in the parametric component of the SVGP:

$$K_{uu}^{-1}m_U = \hat{m}_U \tag{2.59}$$

$$K_{uu}^{-1}S_UK_{uu}^{-1} = \hat{S}_U \tag{2.60}$$

With this parametrization we are left with inverses in the non-parametric part of the SVGP, namely $K_{f_if_i\cdot u} = K_{f_if_i} - K_{f_iu}K_{uu}^{-1}K_{uf_i}$. The authors propose using an upper bound on the latter term:

$$K_{f_if_i\cdot u} = K_{f_if_i} - K_{f_iu}K_{uu}^{-1}K_{uf_i} \leq K_{f_if_i} + K_{f_iu}TK_{uu}TK_{uf_i} - 2K_{f_iu}TK_{uf_i} = \overline{K_{f_if_i\cdot u}} \tag{2.61}$$

with equality only if $T = K_{uu}^{-1}$. We now adapt the lower bound to reflect the change in the non-parametric variance:

$$\mathcal{L}_{IF-LC} = \sum_{i=1}^{N} \mathbb{E}_{\mathcal{N}\left(f_i|\hat{U}(x_i),\hat{\Sigma}(x_i)\right)}\left[\log p(y_i \mid f(x_i))\right] - \mathrm{KL}\left[q(\hat{U})\|p(\hat{U})\right] , \tag{2.62}$$

with $\hat{U}(x_i) = K_{f_iu}\hat{m}_U$ and $\hat{\Sigma}(x_i) = K_{f_if_i} + K_{f_iu}TK_{uu}TK_{uf_i} - 2K_{f_iu}TK_{uf_i} + K_{f_iu}\hat{S}_UK_{uf_i}$. To make sure that this lower bound (which is only valid for log-concave likelihoods) is actually the gap between the true posterior and the approximate posterior, we need to re-frame this lower bound as an SVGP with a structured approximate posterior $q(U^*) = \mathcal{N}\left(U^* \mid m_U^*, S_U^*\right)$, which can be done by matching the posterior generative process in their first two moments, namely:

$$K_{f_iu}K_{uu}^{-1}m_U^* = K_{f_iu}\hat{m}_U \tag{2.63}$$

$$\begin{aligned} K_{f_if_i} - Q_{f_if_i} + K_{f_iu}K_{uu}^{-1}S_U^*K_{uu}^{-1}K_{uf_i} &= K_{f_if_i} + K_{f_iu}TK_{uu}TK_{uf_i} - 2K_{f_iu}TK_{uf_i} \\ &\quad + K_{f_nu}\hat{S}_UK_{uf_n} , \end{aligned} \tag{2.64}$$

where we can see that the solutions are given by $m_U^* = K_{uu}\hat{m}_U$ and $S_U^* = K_{uu} + K_{uu}TK_{uu}TK_{uu} - 2K_{uu}TK_U + K_{uu}\hat{S}_UK_{uu}$. Now we are able to write down the final lower bound for the reparametrized SVGP that is inverse free:

$$\mathcal{L}_{ReSVGP} = \sum_{i=1}^{N} \mathbb{E}_{\mathcal{N}\left(f_i|\hat{U}(x_i),\hat{\Sigma}(x_i)\right)}\left[\log p(y_i \mid f(x_i))\right] - \mathrm{KL}\left[\mathcal{N}\left(m_U^*, S_U^*\right)\|\mathcal{N}\left(0, K_{uu}\right)\right] . \tag{2.65}$$

Throughout the remainder of the thesis, this model will be denoted as Inverse-free Reparametrizable SVPG (IF-ReSVGP).

2.4 Deep Gaussian Processes

Deep Gaussian Processes (DGP) were first introduced in Damianou and Lawrence (2013), as a multi-layered hierarchical formulation of GP. The composition of processes retains the theoretical properties of the underlying stochastic processes (see Kolmogorov consistency theorem) while also ensuring a more diverse hypothesis space of process priors, or at least in theory as we shall later see.

2.4.1 Precursors to Deep Gaussian Processes

The first examples of models where the output of a first GP is used as the input to another second GP can be found in the field of nonlinear time series analysis. A central difficulty in modeling time series data is in determining a model that can capture the nonlinearities of the data without overfitting. GP satisfy these requirements, so foundational work by Wang et al. (2005) introduced the Gaussian Process Dynamical Models (GPDM), which consists of a hierarchical version of the Gaussian Process Latent Variable Model (GPLVM) (Lawrence 2003), where the latent variables are assigned an autoregressive GP prior. This work was generalized in Lawrence and Moore (2007), where the autoregressive modelling assumption is dropped. Both papers assume RBF kernels, a Gaussian likelihood and proceed to use MAP estimates for inference. Subsequent work (Damianou et al. 2011; Titsias and Lawrence 2010) showed how to use sparse variational inference to create variational lower bounds for GPDM and Bayesian GPLVM. Both papers assume a factorized variational posterior across layers.

2.4.2 Probabilistic definition of Deep Gaussian Processes

Building on the the sparse variational framework for integrating out latent variables in Damianou et al. (2011), Damianou and Lawrence (2013) introduced a general model which they coined Deep Gaussian Process (DGP), with added flexibility in choosing the depth and dimensions of hidden layers.

DGP can be interpreted as a cascade of L hidden layers with associated latent variables $\{h_l\}_{l=1}^{L}$ and the observed outputs y are the leaves of the said hierarchy. GP govern the mapping between hidden layers h_l and h_{l+1}, with each layer having a separate GP with specific hyperparameters. Furthermore, each latent variable h_l has additive Gaussian noise $\mathcal{N}(0,\ \sigma_{f,l}^2)$ added.

The joint distribution of a DGP can be written as:

$$p(y, \{h_l\}_{l=1}^{L}) = p(y|h_L) \prod_{l=2}^{L-1} p(h_{l+1}|h_l)\, p(h_1|x) \,, \tag{2.66}$$

where each conditional probability can be expanded to include the GP functional as follows:

$$p(h_{l+1}|h_l) = \int p(h_{h+}|f_{l+1}) p(f_{l+1}|h_l)\, \mathrm{d}f_{l+1} \,. \tag{2.67}$$

To see why the latent variables cannot be integrated out in this construction, we can consider a two layers DGP as:

$$p(h_2) = \int p(h_2|f_2) p(f_2|h_1) p(h_1|h_0)\, \mathrm{d}h_1\, \mathrm{d}f_2 \,, \tag{2.68}$$

which is intractable as the density $p(h_1)$ cannot be propagated through $p(f_2|h_1)$ as it's constructed through a nonlinear covariance function involving h_1.

To circumvent computing this intractable integral, one can make use of the same train of thought as in Damianou et al. (2011), where the introduction of inducing points helps in simplifying approximate inference. Hence, after augmenting only the hidden layers $\{H\}_{l=2}^{L}$, we can now write:

$$p(h_l|U_l, h_{l-1}) = \int p(h_l|f_l) p(f_l|U_l, h_{l-1})\, \mathrm{d}f_l \,, \tag{2.69}$$

where $p(h_l|f_l) = \mathcal{N}\left(h_l \mid f_l, \sigma_{f,l}^2\right)$ and $p(f_l|U_l, h_{l-1})$ is similar to the conditional GP formula introduced in equation (2.17). The authors make the case that since x is deterministic, propagating it through the first GP in the hierarchy is feasible, thus being able to analytically compute $p(h_1|x)$. Therefore, augmenting h_1 with inducing points is not required for the first layer. A graphical model for this construction is given in Figure 2.3a.

The aim is to approximate the log marginal likelihood:

$$\log p(y) = \log \int p(y, \{h_l\}|\{U_l\}) \prod_{l=2}^{L} p(U_l)\, \mathrm{d}\{h_l\}\, \mathrm{d}\{U_l\} \,, \tag{2.70}$$

where we denoted $\{U_l\} = \{U_l\}_{l=2}^{L}$ and $\{h_l\} = \{h_l\}_{l=1}^{L-1}$.

In order to perform VI, the authors introduce a variational distribution:

$$q(U, H) = \prod_{l=2}^{L} q(U_l) \prod_{l=1}^{L} q(h_l) \,, \tag{2.71}$$

which implies a factorisation of the approximate posterior across layers and where the authors pro-

pose parametrizing $q(U_l) = \mathcal{N}(U_l \mid m_{U_l},\ S_{U_l})$ as in the SVGP case. For $\{h_l\}$ the parametrization is split into two cases: $q(h_l) = \prod_{i=1}^{N} \mathcal{N}\left(h_l^{(i)} \mid m_l^{(i)},\ S_l^{(i)}\right)$ for $l \geq 2$ and $q(h_1) = \mathcal{N}(h_1 \mid m_1,\ s_1)$, where S_1 is a full $N \times N$ matrix to account for correlations between data points in the first hidden layer.

By using Jensen's inequality we get a bound $\mathcal{L} \leq \log p(y)$:

$$\mathcal{L} = \int q(U, H) \log \frac{p\left(y, \{h_l\} | \{U_l\}\right) \prod_{l=1}^{L} p(U_l)}{q(U, H)} \, \mathrm{d}\{h_l\} \, \mathrm{d}\{U_l\} \tag{2.72}$$

$$= \int q(U, H) \log p\left(y, \{h_l\} | \{U_l\}\right) \mathrm{d}\{h_l\} \, \mathrm{d}\{U_l\} - \sum_{l=2}^{L} \mathrm{KL}\left[q(U_l) \| p(U_l)\right] + \sum_{l=1}^{L} \mathcal{H}\left[q(h_l)\right] \ , \tag{2.73}$$

where $\mathcal{H}[\cdot]$ denotes the entropy of a distribution and we can expand $\log p\left(y, \{h_l\} | \{u_l\}\right)$ as follows $\log p\left(y | h_l, U_l\right) + \sum_{l=2}^{L-1} \log p\left(h_l | h_{l-1}, U_l\right) + \log p\left(h_1 | x\right)$.

By following the train of derivations introduced in section 2.2.2 we can lower bound $\log p\left(h_l | u_l, h_{l-1}\right)$ by $\log \mathcal{N}\left(h_l | K_{fu} K_{uu}^{-1} U_l, \sigma_{f,l}^2 I_n\right) - \frac{1}{2\sigma_{f,l}^2} \mathrm{Tr}\left[K_{ff} - Q_{ff}\right]$ for $l \geq 2$, where the kernel terms are evaluated using the layer specific kernels, which have different kernel hyperparameters for each layer.

Reintroducing the above into our lower bound we can re-express it as:

$$\mathcal{L} = \sum_{l=2}^{L} \left[\mathbb{E}_{q(h_{l-1}), q(h_l), q(u_l)} \log \mathcal{N}\left(h_l | K_{fu} K_{uu}^{-1} U_l, \sigma_{f,l}^2 I_n\right) - \frac{1}{2\sigma_{f,l}^2} \mathrm{Tr}\left[K_{ff} - Q_{ff}\right] \right] \tag{2.74}$$

$$- \sum_{l=2}^{L} \mathrm{KL}\left[q(U_l) \| p(U_l)\right] + \sum_{l=2}^{L-1} \mathcal{H}\left[q(h_l)\right] - \mathrm{KL}\left[q(h_1) \| p(h_1 | x)\right] \ ,$$

where the KL divergence for h_1 is obtained by combining $\mathcal{H}\left[q(h_1)\right]$ and $\mathbb{E}_{q(h_1)}\left[\log p\left(h_1 | x\right)\right]$.

All of the terms of the above bound are tractable, with easy to compute formulas for the KL and entropy terms. As for the expectation terms for each layer, we refer the reader to Appendix C of Damianou (2015) for explicit analytical equations.

From this variational lower bound we can see that it requires a linearly increasing number of variational parameters with respect to the dataset size, which would pose problems for even medium sized datasets. In the next subsection we shall see how this issue is solved through a different approximate inference procedure.

2.4.3 Doubly Stochastic Variational Inference for Deep Gaussian Processes

A key difference introduced in Salimbeni and Deisenroth (2017) is the removal of hidden layer noise. Hence, we can view the DGP as a composition of functions, keeping in mind that this is only one way of defining this class of probabilistic models (Dunlop et al. 2018):

$$f_L(x) = f_L \circ ... \circ f_1(x) \,, \tag{2.75}$$

with $f_l = \mathcal{GP}\left(m_l, k_l\left(\cdot,\cdot\right)\right)$. Assuming a likelihood function we can write the joint prior as:

$$p\left(y, \{f_l\}_{l=1}^L; X\right) = \underbrace{p(y|f_L)}_{\text{likelihood}} \underbrace{\prod_{l=1}^{L} p(f_l|f_{l-1})}_{\text{prior}} \,, \tag{2.76}$$

with $p\left(f_l \mid f_{l-1}\right) \sim \mathcal{GP}\left(m_l(f_{l-1}), k_l\left(f_{l-1}, f_{l-1}\right)\right)$, where in the case we choose squared exponential kernels we have the following formula for the l-th layer:

$$k^{SE}(f_{l,i}, f_{l,j})_l = \sigma_l^2 \exp\left[\sum_{d=1}^{D_l} -\frac{\left(f_{l,i,d} - f_{l,j,d}\right)^2}{l_{l,d}^2}\right] \,, \tag{2.77}$$

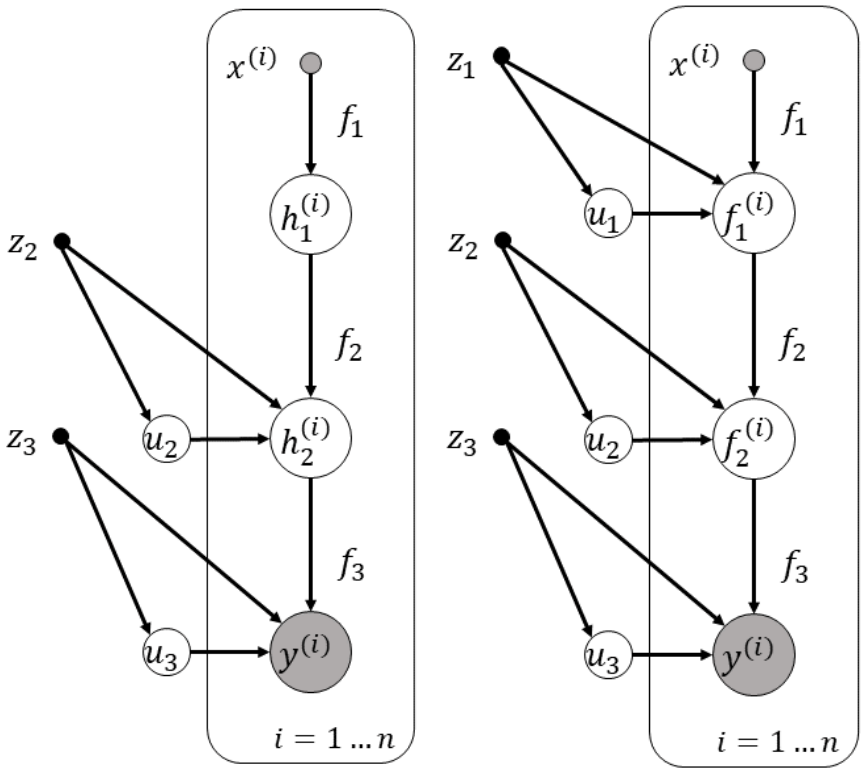
where D_l represents the number of dimensions of F_l and we introduce layer specific kernel hyperparameters $\{\sigma_l^2, l_{l,1}^2, \cdots, l_{l,D_l}^2\}$.

Analytically integrating this Bayesian hierarchical model is intractable as it requires integrating Gaussians which are present in a non-linear way. Moreover, to enable faster inference over our model we can augment each layer l with M_l inducing points' locations Z_{l-1}, respectively inducing points' values U_l resulting in the following augmented joint prior:

$$p\left(y, \{f_l\}_{l=1}^L, \{U_l\}_{l=1}^L; X, \{Z_l\}_{l=0}^{L-1}\right) = \underbrace{p(y|f_L)}_{\text{likelihood}} \underbrace{\prod_{l=1}^{L} p(f_l|f_{l-1}, U_l; Z_{l-1}) p(U_l)}_{\text{prior}} \,, \tag{2.78}$$

where $p(f_l|f_{l-1}, U_l; Z_{l-1}) = \mathcal{N}\left(f_l|m_l(f_{l-1}) + K_{fu}K_{uu}^{-1}\left(U_l - m_l(Z_{l-1}), K_{ff} - K_{fu}K_{uu}^{-1}K_{uf}\right)\right)$. Unlike the training framework introduced in Damianou and Lawrence (2013), we can notice that in this case we augment every hidden layer, including the first one. A graphical model for this new construction is given in Figure 2.3b.

To perform SVI we introduce a factorised variational approximate posterior of the following form: $q\left(\{U_l\}_{l=1}^L\right) = \prod_{l=1}^{L} \mathcal{N}\left(U_l \mid m_{U_l}, S_{U_l}\right)$. Using a similar derivation as in the uncollapsed evidence



(a) DGP (Damianou and Lawrence 2013)

(b) DSVI for DGP (Salimbeni and Deisenroth 2017)

Figure 2.3: Probabilistic graphical models of DGP variants. Left: Original approximate inference framework for training DGP introduced in Damianou and Lawrence (2013); **Right:** DSVI framework for training DGP introduced in Salimbeni and Deisenroth (2017).

lower bound for SVGPs, we can arrive at our ELBO for DGP:

$$\mathcal{L}_{DGP} = \mathbb{E}_{q(\{f_l\}_{l=1}^L)} \left[\log p\left(y \mid f_L\right)\right] - \sum_{l=1}^{L} \mathrm{KL}\left[q(U_l) \| p(U_l)\right] \ , \tag{2.79}$$

where $q\left(\{f_l\}_{l=1}^L\right) = \prod_{l=1}^{L} q\left(f_l \mid f_{l-1}\right)$ and $q\left(f_l \mid f_{l-1}\right) = \mathcal{N}\left(f_l \mid \tilde{U}_l(f_{l-1}), \tilde{\Sigma}_l(f_{l-1})\right)$ and we also have:

$$\tilde{U}_l(f_{l-1}) = m_l(f_{l-1}) + K_{fu}K_{uu}^{-1}\left[m_{U_l} - m_l(Z_{l-1})\right] \tag{2.80}$$

$$\tilde{\Sigma}_l(f_{l-1}) = K_{ff} - K_{fu}K_{uu}^{-1}\left[K_{uu} - S_{U_l}\right]K_{uu}^{-1}K_{uf} \ . \tag{2.81}$$

This composition of functions is approximated via Monte Carlo integration as in the doubly stochastic variational inference (DSVI) framework for training DGP introduced in Salimbeni and Deisenroth (2017).

2.4.4 Advances in approximate inference for Deep Gaussian Processes

Throughout the section we have assumed mean-field approximations between layers. DSVI will remain the method of choice for subsequent chapters in this thesis due to its simplicity and proven results in practice. However, in recent years there has been a steady stream of research delving into improving approximate inference for training DGP.

In Havasi et al. (2018), the authors argue that a drawback of DSVI resides in the fact that it approximates the posterior distribution with a Gaussian, making the case for inference methods that allow a more flexible posterior approximation. With this in mind, the authors propose using Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) (Chen et al. 2014a) for inference. They show that using this framework, they are able to recover multimodal posterior distributions. Under the same motivation, Yu et al. (2019) propose an implicit posterior variational inference (IPVI) framework for DGP that can recover unbiased posterior beliefs while preserving time efficiency. IPVI can be described as a method pertaining to adversarial-based inference (Huszár 2017), producing posterior samples using a black-box generator and dealing with the KL divergence term in the ELBO using the log-density ratio trick. Under the same premise of introducing correlations between the inducing points across layers, Ober and Aitchison (2021) introduced an approximate posterior framework that uses "global" inducing points, which are defined only at the input layer and propagated throughout the hierarchy to obtain inducing inputs in subsequent layers. The authors also propose an alternative framework for obtaining the posterior for inducing points' values at each layer, with the introduction of "inducing-likelihoods", which consist of pseudo-observations and pseudo-precision terms that are combined with the propagated "global" inducing points' values to arrive at an optimal posterior form for inducing points' values. Under this approximate posterior construction, dependencies in $\{U_l\}_{l=1}^L$ are introduced naturally across all layers, implicitly also between $\{f_l\}_{l=1}^L$. This is in stark contrast to the "local" inducing points for each layer and the factorized approximate posterior present in DSVI. A similar approach to the "global" inducing points was also proposed in Ustyuzhaninov et al. (2019), albeit from a different motivation (see next subsection for details). The authors also propose a "jointly Gaussian" approximate posterior for all the layers with analytic formulas for the marginalisation of the "jointly Gaussian" inducing points' values. In the same vein, Lindinger et al. (2020) also proposed structured posteriors across all layers, which they argue helps in obtaining better calibrated uncertainties.

2.4.5 Pathologies in Deep Gaussian Processes

Thus so far we have only approached how to perform approximate inference in DGP. In this subsection we will instead focus on properties of DGP, implicitly also on drawbacks that depend on architecture and approximate inference method of choice.

Feature collapse in DGP: In Duvenaud et al. (2014) the authors show that as you increase the depth of DGP, the sampled functions in prior space will eventually collapse to a small finite set of values. More formally, the authors consider the case of arbitrarily deep, one-dimensional DGP using squared exponential kernels (as defined in equation (2.5)). This choice of kernel implies that its derivative is point-wise distributed as $\mathcal{N}\left(0, \frac{\sigma^2}{l^2}\right)$. Since, we are not interested in the sign of derivatives, we can take their absolute value and recognize it as a half-normal distribution with mean $\sqrt{\frac{2\sigma^2}{\pi l^2}}$. By the chain rule, the absolute value of the derivative of a one-dimensional DGP is the product of the absolute values of derivatives of each layer, which are drawn independently from half-normal distributions. Using the technique of Taylor expansions for the moments of functions of random variables, the authors consider calculating the first two moments of the log of the magnitude of the derivatives, which have the following formulas:

$$m_{log} = \mathbb{E}\left[\log\left|\frac{\partial f(x)}{\partial x}\right|\right] = 2\log\left(\frac{\sigma}{l}\right) - \log 2 - \gamma \tag{2.82}$$

$$var_{log} = \mathbb{V}\left[\log\left|\frac{\partial f(x)}{\partial x}\right|\right] = \frac{\pi^2}{4} + \frac{\log 2}{2} - \gamma^2 - \gamma\log 4 + 2\log\left(\frac{\sigma}{l}\right) \tag{2.83}$$
$$\left[\gamma + \log 2 - \log\left(\frac{\sigma}{l}\right)\right] ,$$

where $\gamma \approx 0.5772$ is the Euler constant. The derivation proceeds by showing that considering the second moment is finite, by the central limit theorem we have:

$$\log\left|\frac{\partial f^{1:L}(x)}{\partial x}\right| = \log\prod_{l=1}^{L}\left|\frac{\partial f^{l}(x)}{\partial x}\right| = \sum_{l=1}^{L}\log\left|\frac{\partial f^{l}(x)}{\partial x}\right| \overset{L\to\infty}{\sim} \mathcal{N}\left(Lm_{log}, L^2 var_{log}\right) . \tag{2.84}$$

Hence, in the limiting case, the distribution of the gradients' size approaches a log-normal distribution. One can see that even though the expected magnitude of the derivative can remain constant, the variance can grow indefinitely as we add more layers to the DGP. The authors argue that since the log-normal distribution is zero lower bounded and heavy-tailed, the vast majority of derivatives will be very small, with extremely rare large jumps.

This phenomenon is shown in Figure 2.4, where we can notice that once the derivative in a specific region of the input space becomes very large or very small, there is a very high chance that it remains like that in subsequent layers. Another observation that can be made regarding this figure is that function values tend to become more restricted to a smaller and smaller set of values as depth increases. Intuitively, this occurs because once function values in different input regions get mapped to similar values in intermediate layers, then given smoothly varying kernels there is no possibility of ever getting mapped to different values later on.

Duvenaud et al. (2014) propose a solution to avert this pathological behaviour, namely that of connecting each intermediate layer of a DGP with the original input space. In the one-dimensional

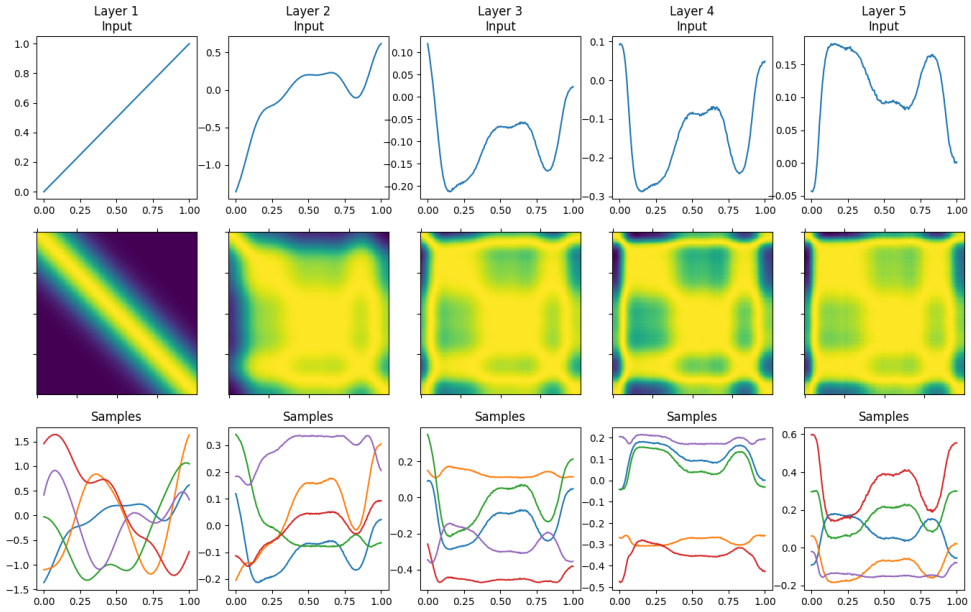


Figure 2.4: Deep Gaussian Processes samples with zero mean function. Top row shows the input to a DGP layer, second row plots the covariance of outputs of that layer and the bottom row shows samples from that layer's output.

DGP scenario, this can be easily achieved by imposing an identity mean function for each GP layer in the hierarchy. This implementation is considered in Figure 2.5, where we can notice that using information from the input space in each hidden layer partially fixes the aforementioned pathology, even though some input space regions can still exhibit small derivatives.

Other research work has since dealt with this pathology in DGP, being mathematically formalized in Dunlop et al. (2018) showing that the function collapse will invariably occur if $\sigma_l^2 < \frac{l_l^2}{D_l}$ is satisfied at each layer in the hierarchy. This suggests that as we increase the width of hidden layers (D_l) in DGP this pathology will disappear.

Compositional uncertainty collapse in DGP: Ustyuzhaninov et al. (2019) considered a yet to be explored research question pertaining to DGP, namely that of compositional uncertainty. The authors make the case that different GP compositions (as part of a DGP) are able to fit the data equally well, thus rendering DGP as inherently unidentifiable. Hence, compositional uncertainty can be viewed as the uncertainty caused by this lack of identifiability. From the uncertainty source decomposition framework introduced in section 1.1, compositional uncertainty can be thought of as the epistemic uncertainty quantifying how the layers jointly compose the data manifold.

In brief, the authors consider a DGP defined as a composition of functions, more precisely $f(x) = (f_L \circ \cdots \circ f_1)(x)$. For each hidden layer, they consider the following alternative expression of the function at layer l expressed as a function of the input stemming from the $l-1$ layer, $f_l(f_{l-1}(x)) =$

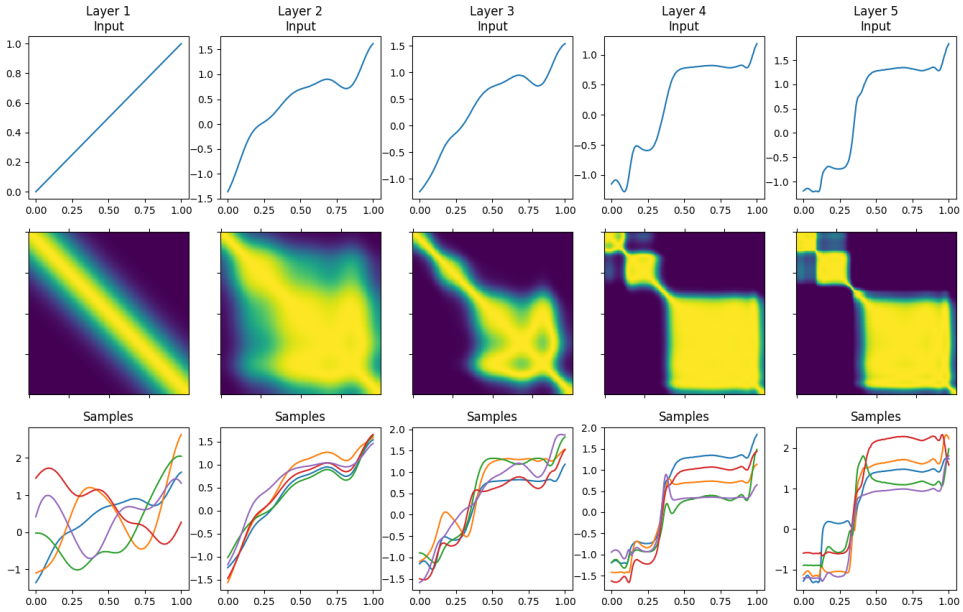


Figure 2.5: Deep Gaussian Processes samples with identity mean function. Top row shows the input to a DGP layer, second row plots the covariance of outputs of that layer and the bottom row shows samples from that layer's output.

$f_l(\mathbb{E}_{f_{l-1}} + \epsilon_{f_{l-1}})$, where $\mathbb{E}_{f_{l-1}}$ represents the expectation of the respective layer and $\epsilon_{f_{l-1}}$ represents a zero-mean noise distribution.

The goal is to show that the variance of f_l increases with increasing variance associated to $\epsilon_{f_{l-1}}$. To see why this is important, consider a noise-free regression task, then given the aforementioned hypothesis, the uncertainty in the hidden layers of the DGP would have to collapse to zero so as to model output data with no noise.

Considering a linear approximation of the form $f_l(f_{l-1}) \approx f_l(\mathbb{E}_{f_{l-1}}) + \epsilon_{f_{l-1}} f_l^{'}(\mathbb{E}_{f_{l-1}})$, we know that a GP and its first order derivative are jointly distributed as follows (McHutchon and Rasmussen 2011):

$$\begin{pmatrix} f_l(\mathbb{E}_{f_{l-1}}) \\ f_l^{'}(\mathbb{E}_{f_{l-1}}) \end{pmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \tilde{U}_l(\mathbb{E}_{f_{l-1}}) \\ \tilde{U}_l^{'}(\mathbb{E}_{f_{l-1}}) \end{bmatrix}, \begin{bmatrix} \tilde{\Sigma}_l(\mathbb{E}_{f_{l-1}}) & \tilde{\Sigma}_l^{'}(\mathbb{E}_{f_{l-1}}) \\ \tilde{\Sigma}_l^{'}(\mathbb{E}_{f_{l-1}}) & \tilde{\Sigma}_l^{''}(\mathbb{E}_{f_{l-1}}) \end{bmatrix} \right) , \tag{2.85}$$

where $\tilde{U}_l(\cdot)$ and $\tilde{\Sigma}(\cdot)$ are defined as in equations (2.80) and (2.81).

We defer the derivation of the following result which stems from the framework of Gaussian approximations of GP with noisy/uncertain inputs (*i.e.*, noise/uncertainty stemming from the previous hidden layer in the hierarchy in this case) to section 2.5. Hence, for a given layer l in a DGP, we can estimate its approximate overall variance as follows:

$$\mathbb{V}\left[f_l\right] = \tilde{\Sigma}_l(\mathbb{E}_{f_{l-1}}) + \mathbb{V}\left[\epsilon_{f_{l-1}}\right]\left[\frac{1}{2}\tilde{\Sigma}_l^{''}(\mathbb{E}_{f_{l-1}}) + \left(\tilde{U}_l^{'}(\mathbb{E}_{f_{l-1}})\right)^2\right] . \tag{2.86}$$

From this formula we can notice that the only term that can be negative is $\tilde{\Sigma}_l^{''}(\mathbb{E}_{f_{l-1}})$, which in turn could cause the approximate variance of this noisy GP to be smaller than the incoming variance (*i.e.*, the overall variance of the previous layer of the DGP denoted $\mathbb{V}\left[\epsilon_{f_{l-1}}\right]$). However, given a good coverage of inducing points across the training set manifold, this term will converge to zero as we increase the number of inducing points. Hence, in practice the probability of obtaining less approximate variance as we progress forward in the hierarchy is almost zero. This proves the fact that for noise-free regression tasks, the variance in the hidden layers will collapse to zero in the case of the DSVI framework for training DGP.

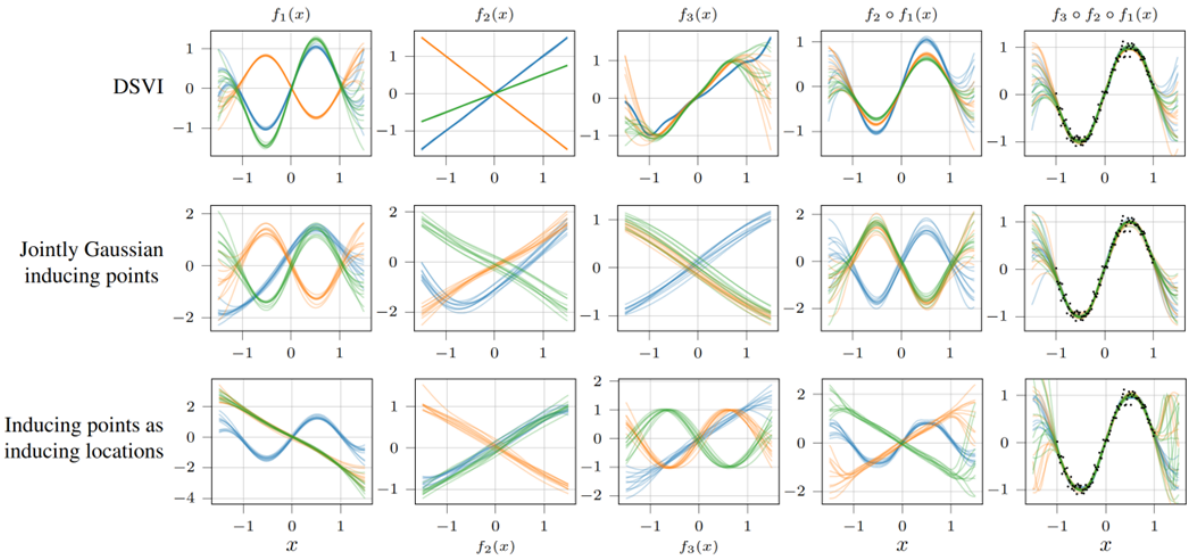


Figure 2.6: Compositional uncertainty collapse in DSVI. Adapted figures from Ustyuzhaninov et al. (2019) showing fits of a three-layer DGP using DSVI as well as variational distributions introduced in Section 4 of Ustyuzhaninov et al. (2019) to a data set shown in the rightmost panel (black dots). Different colours correspond to three models fitted to the same data with different random initialisations. For each initialisation, ten samples (of the same colour) from the fitted model are shown on top of each other.

To solve this pathology, Ustyuzhaninov et al. (2019) consider two alternative approximate posteriors that are correlated between layers. Firstly, a straightforward manner to make the inducing points correlated across layers is to jointly model them with a Gaussian variational distribution, more precisely $q\left(U_1, \cdots, U_L\right) \sim \mathcal{N}\left(m, S\right)$, with $m \in \mathcal{R}^{LM}$ and $S \in \mathcal{R}^{LM \times LM}$. Secondly, they introduce a method entitled "inducing points as inducing locations", which can be succinctly described as follows: instead of explicitly optimizing over inducing locations $\{Z_l\}_{l=1}^{L}$, there is just one set of inducing locations Z operating in the input space, with subsequent realisations of inducing locations in hidden layers being obtained by passing the previous ones through the GP governing the mapping between layers. Essentially, the inducing points of a previous layer are the inducing lo-

cations of the next one. The rationale behind these approximate inference methods that introduce correlations between layers is that factorised approximate posteriors between layers will force the GP mappings between layers to be deterministic to as to obtain a posterior in observation space that is noise-free (see Figure 1 in Ustyuzhaninov et al. (2019) for a conceptual illustration of this phenomenon).

In Figure 2.6 we can notice that for DGP trained on noise-free regression tasks, the DSVI method essentially collapses to almost deterministic transformations in the hidden layers. The remedies suggested by the authors seem to increase the uncertainty since the approximate posteriors are correlated across layers.

2.5 Approximation of Gaussian Processes with noisy inputs

As we have seen in the previous subsection, measuring uncertainties in the hidden layers of DGP is important so as to avoid pathological behaviour. In order to analyse these properties and to establish necessary conditions for the reliable propagation of uncertainty in DGP, we need a mathematical framework with analytical equations (such as the previously introduced equation (2.86)). In this section we lay out this precise toolkit, keeping in mind that these theoretical results will play a key role in Chapter 3.

In section 2.1 we have seen that the predictive distribution of a GP at a unseen noise-free input is Gaussian. In Girard (2004), the authors propose several techniques to approximate the non-Gaussian predictive distribution of a GP when the input is noisy (or a random variable in a more general scenario as we shall later see). In this section we will attempt to make a summary of aforementioned work, keeping in mind that these approximation techniques will play a key role in analyzing properties of DGP and how they propagate uncertainty through the hierarchy, since we can view the hidden layers of a DGP as GP with an uncertain input.

We now consider the case where the input is corrupted by noise $\epsilon_{x^*} \sim \mathcal{N}(0, \Sigma_{x^*})$ such that $x^* = \mu_{x^*} + \epsilon_{x^*}$. Hence, our aim is to make predictions at $x^* \sim \mathcal{N}(\mu_{x^*}, \Sigma_{x^*})$, which involves the following integral:

$$p(y^* \mid D, \mu_{x^*}, \Sigma_{x^*}) = \int p(y^* \mid D, x^*)\, p(x^* \mid \mu_{x^*}, \Sigma_{x^*})\, \mathrm{d}x^*, \tag{2.87}$$

where $p(y^* \mid D, x^*) = \frac{1}{\sqrt{2\pi\sigma^2(x^*)}} \exp\left[-\frac{1}{2}\frac{(y^*-\mu(x^*))^2}{\sigma^2(x^*)}\right]$, $\mu(x^*) = K_{f^*f}K_{ff}^{-1}y$ and $\sigma^2(x^*) = K_{f^*f^*} - Q_{f^*f^*}$. In this subsection, for notation purposes, we will consider $K_{x_i,x_j} = K^{SE}(x_i, x_j) + \delta_{i,j}\sigma^2_{noise}$. One can notice that $p(y^* \mid D, x^*)$ is a non-linear function of x^*, so we cannot integrate out x^*. We will now consider approximations to the solution of this integral.

I Monte Carlo Integration: uses random sampling of a function to numerically compute an estimate of its integral.

$$p^{MC}\left(y^* \mid D, \mu_{x^*}, \Sigma_{x^*}\right) \approx \frac{1}{T}\sum_{t=1}^{T} p\left(y^* \mid D, x_t^*\right) = \frac{1}{T}\sum_{t=1}^{T} \mathcal{N}\left(y^* \mid \mu(x^*), \sigma^2(x^*)\right) \ . \tag{2.88}$$

It will result in a mixture of T Gaussian distributions with equal mixing weights and as $T \to \infty$ we have $p^{MC}\left(y^* \mid D, \mu_{x^*}, \Sigma_{x^*}\right) \approx p\left(y^* \mid D, \mu_{x^*}, \Sigma_{x^*}\right)$. We remind ourselves that the case $T = 1$ is the default method to computing this type of integrals within the framework of DSVI for training DGP.

II Gaussian Approximation: involves approximating the non-Gaussian predictive equation with a Gaussian distribution:

$$p\left(y^* \mid D, \mu_{x^*}, \Sigma_{x^*}\right) = \mathcal{N}\left(y^* \mid m(u_{x^*}, \Sigma_{x^*}), v(u_{x^*}, \Sigma_{x^*})\right) \ . \tag{2.89}$$

We can now explicitly write the first two moments:

$$\begin{aligned}
m(u_{x^*}, \Sigma_{x^*}) &= \int y^* \left[\int p\left(y^* \mid D, x^*\right) p\left(x^* \mid u_{x^*}, \Sigma_{x^*}\right) dx^*\right] dy^* && (2.90)\\
&= \int \left[\int y^* p\left(y^* \mid D, x^*\right) dy^*\right] p\left(x^* \mid u_{x^*}, \Sigma_{x^*}\right) dx^* && (2.91)\\
&= \int \mu(x^*) p\left(x^* \mid u_{x^*}, \Sigma_{x^*}\right) dx^* && (2.92)\\
&= \mathbb{E}_{x^*}\left[\mu(x^*)\right] \ . && (2.93)
\end{aligned}$$

$$\begin{aligned}
v(u_{x^*}, \Sigma_{x^*}) &= \int (y^*)^2 \left[\int p\left(y^* \mid D, x^*\right) p\left(x^* \mid u_{x^*}, \Sigma_{x^*}\right) dx^*\right] dy^* - m(u_{x^*}, \Sigma_{x^*})^2 && (2.94)\\
&= \int \left[\int (y^*)^2 p\left(y^* \mid D, x^*\right) dy^*\right] p\left(x^* \mid u_{x^*}, \Sigma_{x^*}\right) dx^* - m(u_{x^*}, \Sigma_{x^*})^2 && (2.95)\\
&= \int \left[\sigma^2(x^*) + \mu(x^*)^2\right] p\left(x^* \mid u_{x^*}, \Sigma_{x^*}\right) dx^* - m(u_{x^*}, \Sigma_{x^*})^2 && (2.96)\\
&= \mathbb{E}_{x^*}\left[\sigma^2(x^*)\right] + \mathbb{E}_{x^*}\left[\mu(x^*)^2\right] - m(u_{x^*}, \Sigma_{x^*})^2 && (2.97)\\
&= \mathbb{E}_{x^*}\left[\sigma^2(x^*)\right] + \mathbb{V}_{x^*}\left[\mu(x^*)\right] \ , && (2.98)
\end{aligned}$$

, where we now consider just predictive distributions over a single data point x^*, hence we update predictive equations to correspond to the univariate prediction case:

$$\mu(x^*) = K_{x^*f}K_{ff}^{-1}y \tag{2.99}$$

$$\sigma^2(x^*) = K_{x^*x^*} - \sum_{i,j=1}^{N} K_{ij}^{-1}K(x^*, x_i)K(x^*, x_j)\ . \tag{2.100}$$

To recap, we are interested in finding the first two moments of the Gaussian approximation in equation (2.89), which we now write explicitly:

$$m(u_{x^*}, \Sigma_{x^*}) = \mathbb{E}_{x^*}\left[K_{x^*f}K_{ff}^{-1}y\right] \tag{2.101}$$

$$\begin{aligned} v(u_{x^*}, \Sigma_{x^*}) = \mathbb{E}_{x^*}&\left[K_{x^*x^*} - \sum_{i,j=1}^{N} K_{ij}^{-1}K(x^*, x_i)K(x^*, x_j)\right] \\ &+ \mathbb{E}_{x^*}\left[\left[K_{ff}^{-1}y\right]_{i,:}\left[K_{ff}^{-1}y\right]_{j,:}K_{x^*,x_i}K_{x^*,x_j}\right] \\ &- m(u_{x^*}, \Sigma_{x^*})^2\ , \end{aligned} \tag{2.102}$$

where $\left[K_{ff}^{-1}y\right]_{i,:}$ takes the i-th row. Therefore, we need to compute the following integrals:

$$\Psi = \mathbb{E}_{x^*}\left[K_{x^*,x^*}\right] = \int K(x^*, x^*)p(x^*)\,\mathrm{d}x^* \tag{2.103}$$

$$\Psi_i = \mathbb{E}_{x^*}\left[K_{x^*,x_i}\right] = \int K(x^*, x_i)p(x^*)\,\mathrm{d}x^* \tag{2.104}$$

$$\Psi_{i,j} = \mathbb{E}_{x^*}\left[K_{x^*,x_i}K_{x^*,x_j}\right] = \int K(x^*, x_i)K(x^*, x_j)p(x^*)\,\mathrm{d}x^*\ . \tag{2.105}$$

These integrals will have an analytic solution in the case of specific kernels. We will first focus on the general case, whereby we devise approximations for any arbitrary kernel, subsequently calculating the exact moments for squared exponential kernels, which are the default kernels used in DGP throughout the rest of this thesis.

2.5.1 Approximate Moments

Moment approximations involves approximating the integrand by a Taylor expansion.

General case. To illustrate this method, we take a simple scenario where x is a random variable such that $\mathbb{E}_x\left[x\right] = \mu_x$ and $\mathbb{V}_x\left[x\right] = \Sigma_x$. For a well-behaved function f with $y = f(x)$ and sufficiently small $\sigma_x = \sqrt{\Sigma_x}$ we can consider the Taylor expansion of $f(x)$ at μ_x up to second

order to be an accurate representation of $f(x)$:

$$y = f(x) \approx f(\mu_x) + (x - \mu_x) f^{'}(\mu_x) + \frac{1}{2}(x - \mu_x)^2 f^{''}(\mu_x) \tag{2.106}$$

Using the Taylor expansion, we can calculate the first two moments as follows:

$$\mathbb{E}_x[y] \approx \mathbb{E}_x[f(\mu_x)] + \mathbb{E}_x\left[(x - \mu_x) f^{'}(\mu_x)\right] + \mathbb{E}_x\left[\frac{1}{2}(x - \mu_x)^2 f^{''}(\mu_x)\right] \tag{2.107}$$

$$\approx f(\mu_x) + \frac{1}{2}\Sigma_x f^{''}(\mu_x) \tag{2.108}$$

$$\mathbb{V}_x[y] \approx \mathbb{E}_x\left[y^2\right] - \mathbb{E}_x[y]^2 \tag{2.109}$$

$$\approx \underbrace{f(\mu_x)^2 + \Sigma_x f^{'}(\mu_x)^2 + f(\mu_x) f^{''}(\mu_x)\Sigma_x}_{\mathbb{E}_x[y^2]} - \underbrace{f(\mu_x)^2 - f(\mu_x) f^{''}(\mu_x)\Sigma_x}_{\mathbb{E}_x[y]^2} \tag{2.110}$$

$$\approx \Sigma_x f^{'}(\mu_x)^2 \tag{2.111}$$

GP with noisy inputs. We now apply the above results, more specifically equations (2.108) and (2.111), to equations (2.93) and (2.98), obtaining:

$$m(u_{x^*}, \Sigma_{x^*}) = \mathbb{E}_{x^*}[\mu(x^*)] \tag{2.112}$$

$$= \mu(\mathbb{E}_{x^*}(x^*)) + \frac{1}{2}\mathbb{V}_{x^*}(x^*)\mu^{''}(\mathbb{E}_{x^*}[x^*]), \tag{2.113}$$

where we have considered a Taylor expansion of $\mu(x^*)$ and used the first moment as given by the formula in equation (2.108).

In a similar manner, we can also derive the second moment:

$$v(u_{x^*}, \Sigma_{x^*}) = \mathbb{E}_{x^*}\left[\sigma^2(x^*)\right] + \mathbb{V}_{x^*}[\mu(x^*)] \tag{2.114}$$

$$= \underbrace{\sigma^2(\mathbb{E}_{x^*}[x^*]) + \mathbb{V}_{x^*}\frac{1}{2}\sigma^2(\mathbb{E}_{x^*}[x^*])^{''}}_{\mathbb{E}_{x^*}[\sigma^2(x^*)]} + \underbrace{\mathbb{V}_{x^*}\mu^{'}(\mathbb{E}_{x^*}[x^*])^2}_{\mathbb{V}_{x^*}[\mu(x^*)]} \tag{2.115}$$

$$= \sigma^2(\mathbb{E}_{x^*}[x^*]) + \mathbb{V}_{x^*}\left[\frac{1}{2}\sigma^2(\mathbb{E}_{x^*}[x^*])^{''} + \mu^{'}(\mathbb{E}_{x^*}[x^*])^2\right], \tag{2.116}$$

where we have considered Taylor expansions of $\sigma^2(x^*)$ and $\mu(x^*)$ alongside using either the first or second moments as given by the formulas in equations (2.108) and (2.111) pending on case.

2.5.2 Exact Moments

We can calculate the exact moments in the case of squared exponential kernels of the following form:

$$K(x_i, x_j) = \sigma^2 \exp\left[-\frac{1}{2}[x_i - x_j]^\top W^{-1}[x_i - x_j]\right], \tag{2.117}$$

where we usually consider just $W = \text{diag}\left[l_1^2, \cdots, l_D^2\right]$.

We remind ourselves that we need to compute the following equations:

$$m\left(\mu_{x^*}, \Sigma_{x^*}\right) = \sum_{i=1}^{N} \left[yK_{ff}^{-1}\right]_{i,:} \Psi_i \tag{2.118}$$

$$v\left(\mu_{x^*}, \Sigma_{x^*}\right) = \Psi - \sum_{i,j=1}^{N} \left[K_{ij}^{-1} - \left[yK_{ff}^{-1}\right]_{i,:} \left[yK_{ff}^{-1}\right]_{j:}\right] \Psi_{ij} - m\left(\mu_{x^*}, \Sigma_{x^*}\right)^2 \ . \tag{2.119}$$

We now only need to derive the analytic formulas for Ψ, Ψ_i and Ψ_{ij}, all of which will make use of the following identity.

Definition 2.5.1 (Product of Multivariate Gaussians)**.** For two multivariate Gaussian distributions, $\mathcal{N}\left(x \mid a, A\right)$ and $\mathcal{N}\left(x \mid b, B\right)$, their product can be expressed as follows:

$$zN\left(x \mid c, C\right) = \mathcal{N}\left(x \mid a, A\right)\mathcal{N}\left(x \mid b, B\right), \tag{2.120}$$

where $c = C\left[A^{-1}a + B^{-1}b\right]$ and $C = \left[A^{-1} + B^{-1}\right]^{-1}$.

Armed with this toolkit we can proceed further:

$$\Psi = \mathbb{E}_{x^*}\left[K\left(x^*, x^*\right)\right] = \sigma^2 \tag{2.121}$$

$$\Psi_i = \mathbb{E}_{x^*}\left[K\left(x^*, x_i\right)\right] \tag{2.122}$$

$$= \int K\left(x^*, x_i\right)\mathcal{N}\left(x^* \mid \mu_{x^*}, \Sigma_{x^*}\right)\mathrm{d}x^* \tag{2.123}$$

$$= c\int \mathcal{N}\left(x^* \mid x^i, W\right)\mathcal{N}\left(x^* \mid \mu_{x^*}, \Sigma_{x^*}\right)\mathrm{d}x^* \tag{2.124}$$

$$= c\mathcal{N}\left(\mu_{x^*} \mid x_i, W + \Sigma_{x^*}\right) \tag{2.125}$$

$$\Psi_{ij} = \mathbb{E}_{x^*}\left[K\left(x^*, x_i\right)K\left(x^*, x_j\right)\right] \tag{2.126}$$

$$= c^2\int \mathcal{N}\left(x^* \mid x^i, W\right)\mathcal{N}\left(x^* \mid x^j, W\right)\mathcal{N}\left(x^* \mid \mu_{x^*}, \Sigma_{x^*}\right)\mathrm{d}x^* \tag{2.127}$$

$$= c^2\mathcal{N}\left(x_i \mid x_j, 2W\right)\int \mathcal{N}\left(x^* \mid \frac{x_i + x_j}{2}, \frac{W}{2}\right)\mathcal{N}\left(x^* \mid \mu_{x^*}, \Sigma_{x^*}\right)\mathrm{d}x^* \tag{2.128}$$

$$= c^2\mathcal{N}\left(x_i \mid x_j, 2W\right)\mathcal{N}\left(\mu_{x^*} \mid \frac{x_i + x_j}{2}, \Sigma_{x^*} + \frac{W}{2}\right) \ , \tag{2.129}$$

where $c = (2\pi)^{\frac{D}{2}}\left|W\right|^{\frac{1}{2}}\sigma^2$. We can now write the full explicit form for the first two exact mo-

ments:

$$m\left(\mu_{x^*}, \Sigma_{x^*}\right) = \sum_{i=1}^{N} \left[yK_{ff}^{-1}\right]_{i,:} c\mathcal{N}\left(\mu(x^*) \mid x_i, W + \Sigma_{x^*}\right) \tag{2.130}$$

$$v\left(\mu_{x^*}, \Sigma_{x^*}\right) = \sigma^2 - c^2 \sum_{i,j=1}^{N} \left[yK_{ff}^{-1}\right]_{i,:} \left[yK_{ff}^{-1}\right]_{j,:} \mathcal{N}\left(x_i \mid x_j, 2W\right) \tag{2.131}$$
$$\mathcal{N}\left(\mu_{x^*} \mid \frac{x_i + x_j}{2}, \Sigma_{x^*} + \frac{W}{2}\right) - m\left(\mu_{x^*}, \Sigma_{x^*}\right)^2 \ .$$

Following Quinonero-Candela et al. (2003) we can re-write the explicit first two moments as:

$$m\left(\mu_{x^*}, \Sigma_{x^*}\right) = q^\top K_{ff}^{-1} y \tag{2.132}$$

$$v\left(\mu_{x^*}, \Sigma_{x^*}\right) = \sigma^2 + Tr\left[\left(K_{ff}^{-1} y y^\top K_{ff}^{-\top} - K_{ff}^{-1}\right) Q\right] - \mathrm{Tr}\left[y^\top K_{ff}^{-1} y\right] \ , \tag{2.133}$$

where we have:

$$q_i = \left|W^{-1}\Sigma_{x^*} + I_n\right|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\left(\mu_{x^*} - x_i\right)^\top \left(\Sigma_{x^*} + W\right)^{-1} \left(\mu_{x^*} - x_i\right)\right] \tag{2.134}$$

$$Q_{ij} = \left|W^{-1}\Sigma_{x^*} + I_n\right|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\left(\frac{x_i + x_j}{2} - \mu_{x^*}\right)^\top \left(\Sigma_{x^*} + \frac{1}{2}W\right)^{-1} \left(\frac{x_i + x_j}{2} - \mu_{x^*}\right)\right] \tag{2.135}$$
$$\exp\left[-\frac{1}{2}\left(x_i - x_j\right)^\top \left(2W\right)^{-1} \left(x_i - x_j\right)\right] \ .$$

As we shall later see in Chapter 3, these results will play a key role in analysing how uncertainty propagates in the hidden layers of DGP.

Chapter 3

Distributional Deep Gaussian Processes

This chapter is based on the following publication:

(a) **Popescu, S. G.**, Sharp, D., Cole, J., and Glocker, B. (2020). "Hierarchical Gaussian Processes with Wasserstein-2 Kernels". In: *NeurIPS Workshop on Bayesian Deep Learning* – (Popescu et al. 2020)

The code for all experiments is available at https://github.com/SebastianPopescu/DistDGPs.

Stacking Gaussian Processes severely diminishes the model's ability to detect outliers, which when combined with non-zero mean functions, further extrapolates low non-parametric variance to low training data density regions. This phenomenon is a direct product of Euclidean space kernels not taking into account the geometric information available in the hidden layers of the hierarchy. We propose a hybrid kernel inspired from Varifold theory, operating in both Euclidean and Wasserstein space. We posit that directly taking into account the variance in the computation of Wasserstein-2 distances is of key importance towards maintaining outlier status throughout the hierarchy. We show improved performance on medium and large scale datasets and enhanced out-of-distribution detection on both toy and real data.

3.1 Introduction

GP are data-efficient Bayesian non-parametric models that offer calibrated uncertainty quantification and are robust to overfitting, recently finding applicability in data-scarce domains (Timonen et al. 2019; Wang et al. 2020) or where uncertainty is of utmost importance (Chen et al. 2014b). DGP

(Damianou and Lawrence 2013) are multi-layered generalizations of GP that share data-efficiency learning capabilities and show enhanced representational capacity. However, the reliability of uncertainties propagated throughout the hierarchy is up for debate. Recent work (Ustyuzhaninov et al. 2019) has questioned the uncertainties present in the hidden layers of DGP, showing that approximate inference schemes using factorized variational Gaussian distributions result in all but the last GP collapsing to deterministic transformations in the case of noise-free data. Such pathological behaviour should be avoided as it undermines the utility of multi-layered GP. We propose desiderata for useful properties that a kernel must have in a DGP and based on this we introduce an alternative kernel operating in both Euclidean and Wasserstein-2 space, showcasing that taking into consideration the explicit distance between multivariate Gaussians present in hidden layers is of vital importance towards correctly propagating non-parametric uncertainty forward. Prior work (Bachoc et al. 2017; Bachoc et al. 2018; Thi Thien Trang et al. 2019) has only focused on shallow GP operating on inputs which do not have an explicit distribution, thus using Wasserstein-2 based kernels to take advantage of the distributional nature of the input. We did not find any other work that expanded the concept of Wasserstein-2 based kernels to DGP.

Contributions To summarize, this chapter makes the following advancements:

- We provide empirical evidence that as the number of inducing points increases, the non-parametric variance (*i.e.*, variance of Schur complement in conditioned GP) in hidden layers progressively collapses in the case of zero-mean function DGP.
- We show that using Principal Component Analysis (PCA) mean functions solves this problem, but this procedure induces low variance perpendicular to the eigenvectors in the case of encoding down high-dimensional data.
- We derive theoretical requirements for DGP to propagate outliers with high total uncertainty, respectively high non-parametric variance (*i.e.*, variance of Schur complement in conditioned GP) throughout the hierarchy.
- We propose a variant of DGP that operates on both Euclidean and Wasserstein-2 space. We demonstrate that this formulation guarantees higher non-parametric variance for OOD points compared to in-distribution points, alongside increased chances of not suffering from *feature-collapse*.
- We empirically show enhanced OOD detection capabilities, alongside improved test time results on a wide range of datasets for our proposed model.

3.2 Background

In this section we introduce the necessary theoretical background needed for this chapter.

3.2.1 Wasserstein-2 distance kernels on Gaussian measures

The Wasserstein space on $\mathbb{R}$ can be defined as the set $W_2(\mathbb{R})$ of probability measures on $\mathbb{R}$ with a finite second moment. We denote by $\Pi(\mu,\nu)$ the set of all probability measures Π over the product set $\mathbb{R}\times\mathbb{R}$ with marginals μ and ν, which are probability distributions in $W_2(\mathbb{R})$. The transportation cost between two measures μ and ν is defined as $T_2(\mu,\nu)=\inf_{\pi\in\Pi(\mu,\nu)}\int[x-y]^2\mathrm{d}\pi(x,y)$. This transportation cost allows us to endow the set $W_2(\mathbb{R})$ with a metric by defining the quadratic Wasserstein distance between μ and ν as $W_2(\mu,\nu)=T_2(\mu,\nu)^{1/2}$. More details on Wasserstein spaces and their links with optimal transport problems can be found in Villani (2008).

Theorem 3.2.1 (Theorem IV.1. in Bachoc et al. 2017)**.** *Let $k_W : W_2(\mathbb{R})\times W_2(\mathbb{R})\to\mathbb{R}$ be the Wasserstein-2 RBF kernel defined as following:*

$$k^{W_2}(\mu,\nu)=\sigma^2\exp\frac{-W_2^2(\mu,\nu)}{l^2}=\sigma^2\exp\frac{-T_2(\mu,\nu)}{l^2}\,, \tag{3.1}$$

then $k^{W_2}(\mu,\nu)$ is a positive definite kernel for any $\mu,\nu\in W_2(\mathbb{R})$, respectively σ^2 is the kernel variance, l^2 being the lengthscale.

For completeness a proof following the layout in Thi Thien Trang et al. (2019) is provided in Appendix A.2.1.

Multiplication of positive definite kernels results again in a positive definite kernel, hence we arrive at the automatic relevance determination kernel based on Wasserstein-2 distances:

$$k^{W_2}([\mu_d]_{d=1}^D,[\nu_d]_{d=1}^D)=\sigma^2\exp\sum_{d=1}^D\frac{-W_2^2(\mu_d,\nu_d)}{l_d^2}\,. \tag{3.2}$$

Wasserstein-2 Distance between Gaussian distributions Gaussian measures fulfill the condition of finite second moment, thereby being a clear example of probability measures for which we can compute Wasserstein metrics. The Wasserstein-2 distance between two multivariate Gaussian distributions $\mathcal{N}(m_1,\Sigma_1)$ and $\mathcal{N}(m_2,\Sigma_2)$, which have associated Gaussian measures and implicitly the Wasserstein metric is well defined for them, has been shown to have the following form $\parallel m_1-m_2\parallel_2^2+\mathrm{Tr}\left[\Sigma_1+\Sigma_2-2\left(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2}\right)^{1/2}\right]$ (Dowson and Landau 1982), which in the case of univariate Gaussians simplifies to $|m_1-m_2|^2+|\sqrt{\Sigma_1}-\sqrt{\Sigma_2}|^2$. This last formulation will be used throughout this paper.

3.2.2 Varifold theory or kernels between generalized measure spaces

Subsequently introduced kernels in this chapter were inspired by Varifold kernels, which succinctly can be described as a composite kernel for shape matching (Kaltenmark et al. 2017) between manifolds that includes an Euclidean part and also a part that takes into account geometrical in-

formation as given by the tangent space. A brief mathematical introduction to Varifold theory is given in Appendix A.1.

An illustration of a varifold kernel is given in Figure 3.1. Concisely, varifold kernels use the distance between Euclidean points on surface meshes but also a distance based on their surface normal vectors. This provides us with a hint towards better construction of kernels for DGP (*i.e.*, hybrid kernel from Figure 3.1 will be introduced in section 3.4.2), ones that also take into account geometric information of the underlying mixture of Gaussians distributions. Wasserstein distances (Kolouri et al. 2017) are well studied on Gaussian distributions (Mallasto and Feragen 2017; Takatsu 2011), making them a prime candidate for developing kernels operating on distributions.

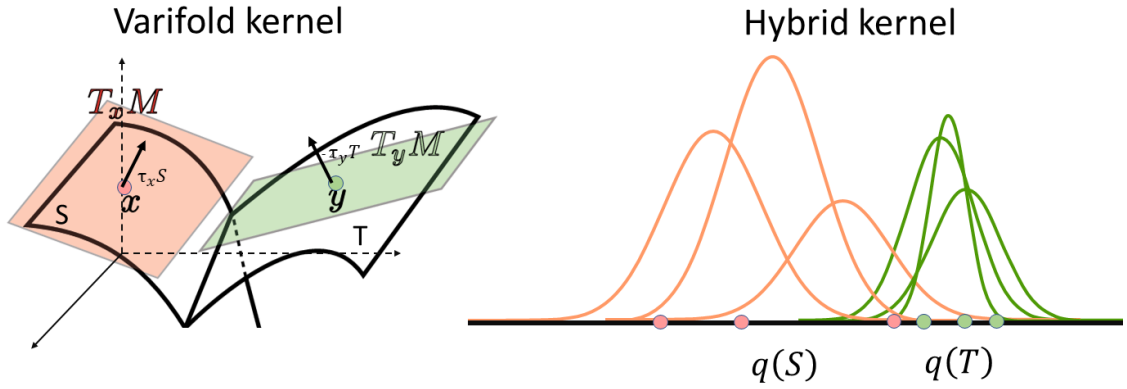


Figure 3.1: Similarities between Varifold kernel and Hybrid kernel. Left: Comparing two manifolds involves taking 3D points in Euclidean space (denoted by circles) and their associated surface normal vectors (denoted by directed arrows), subsequently computing their distance in their respective domains. We only show the comparison between two points for aesthetical reasons, however it should be noted that in practice a multitude of points and associated surface normal vectors are used; **Right:** Considering two data points S and T, their predictive distribution at a certain hidden layer of a DGP would consist of mixture of Gaussians (p.d.f. of mixture components are depicted either in pink or green). The hybrid kernel takes samples from mixture components and their underlying Gaussian distributions into consideration, thereby using geometric information.

3.3 Propagation of variance in DGP

Ustyuzhaninov et al. (2019) have proved for noise-free data that the various stochastic layers in the hierarchy of a DGP collapse to deterministic transformations. In this section, we analyze what are the necessary theoretical requirements for a zero-mean DGP to maintain comparatively higher total uncertainty for an OOD data point throughout the hierarchy and draw some insights into what type of uncertainty (*e.g.*, parametric/epistemic or non-parametric/distributional) is propagated forward in each case. We analyze two different case studies that might occur in practice. The first one involves inducing points $\{Z_0, \cdots, Z_L\}$ that are evenly spread around 0 with a maximum range of $[-3\sigma_l, 3\sigma_l]$ at each layer such that samples stemming from the previous layers are within the neighbourhood of said inducing points. Secondly, we will consider the scenario where we have

two separate groups of inducing points $Z_{l,1} \subsetneq [-\infty, -c]$ and $Z_{l,2} \subsetneq [c, \infty]$ for $c \in \mathbb{R}^+$. This situation can occur in practice for a binary classification task, where at the respective layer, the inducing points can reliably separate the two classes in hidden layer feature space. We now detail what are the necessary conditions in the first scenario to predict higher total uncertainty for OOD data points in comparison to in-distribution points.

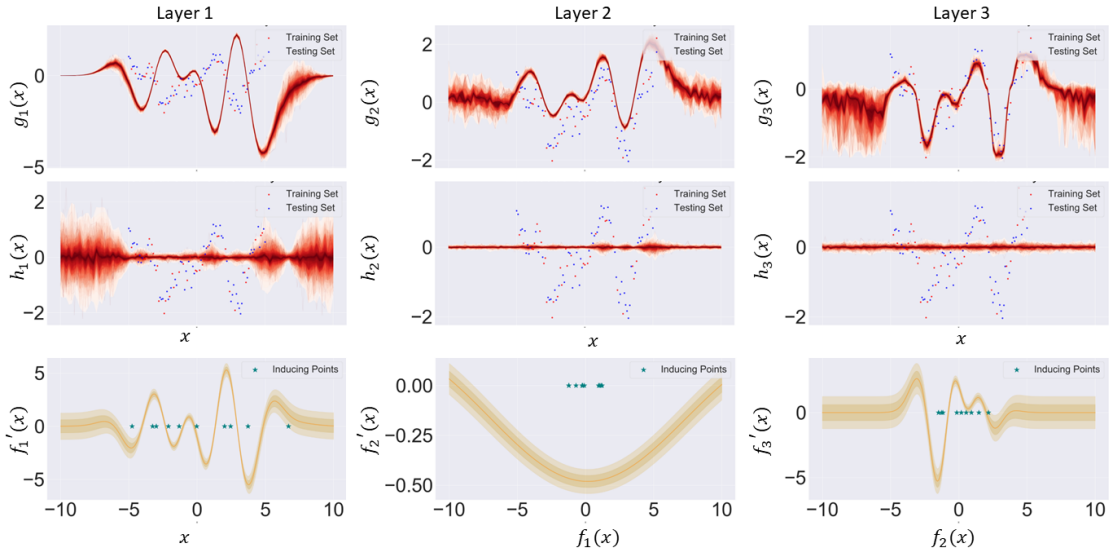


Figure 3.2: Uncertainty in DGP is conditioned on first order derivatives. Top row: Layer-wise parametric component $G_l(x)$ showing epistemic uncertainty; **Mid row:** Layer-wise non-parametric component $H_l(x)$ showing distributional uncertainty. We can observe that for the hidden layers it collapses to almost zero, thus rendering $H_l(x)$ almost useless except for the first layer; **Bottom row:** Layer-wise first order derivative $f_l^{'}(f_{l-1}(x))$ showing regions with large derivatives. Since the maximum absolute derivative values are reached in regions where OOD points from the previous layer are mapped, it has the effect of producing high epistemic uncertainty in OOD regions (see regions outside data manifold in top row).

Proposition 3.3.1. *We consider the approximate posterior DGP as a composition of functions* $q_L = q_L \circ ... \circ q_1$ *with* q_l *being given by* $q(F_l|F_{l-1}) = \mathcal{N}(F_l \mid \tilde{U}_l(F_{l-1}), \tilde{\Sigma}_l(F_{l-1}))$. *We consider an OOD data point* x_{out} *in input space such that* $\tilde{\Sigma}_1(x_{out}) = \sigma_1^2$ *and* $\tilde{U}_1(x_{out}) = 0$, x_{in-d} *to be in-distribution, with* $\tilde{\Sigma}_1(x_{in-d}) = V_{in-d} \leq \sigma_1^2$ *and* $\tilde{U}_1(x_{in-d}) = M_{in-d}$. *We assume that* Z_2 *are equidistantly placed between* $[-3\sigma_1, 3\sigma_1]$. *The variance in the second layer of* x_{out} *will be higher than* x_{in-d} *if the following holds:*

$$\frac{\left(\frac{\partial \tilde{U}_2(F_1)}{\partial F_1}\right)^2\Big|_{F_1=M_{in-d}}}{\left(\frac{\partial \tilde{U}_2(F_1)}{\partial F_1}\right)^2\Big|_{F_1=0}} \leq \frac{\sigma_1^2}{V_{in-d}} \,. \tag{3.3}$$

A proof of this proposition can be found in Appendix A.2.2.

Remark 3.3.1. *Intuitively, this inequality can be easily satisfied if* $\left(\frac{\partial \tilde{U}_2(F_1)}{\partial F_1}\right)^2\Big|_{F_1=0}$ *reaches its abso-*

lute peak at $F_1 = 0$ since $1 \leq \frac{\sigma_1^2}{V_{in-d}}$ holds in general. On a toy regression task we can observe that the overall uncertainty is correctly predicted as higher for OOD points due to high absolute values of the derivative at 0 (i.e., where OOD points get mapped in hidden layers on average) in hidden layers' space (see top and bottom rows of Figure 3.2). However, this requirement for correctly propagating total uncertainty is not guaranteed to be satisfied in practice in all scenarios.

In the first investigated case, we have thus established the necessary condition for the reliable propagation of total variance so as to guarantee that OOD data points have higher total uncertainty compared to in-distribution data points at each layer in the hierarchy. However, a reasonable question is whether DGP in this case are *distance-aware* (see Definition 1.2.1) at each layer? From Figure 3.2 we can see that the non-parametric uncertainty (which stands as a proxy for distributional uncertainty; the two terms will be used interchangeably throughout the remainder of the thesis) collapses to zero across the whole spectrum of data points in the hidden layers. Therefore, DGP in this scenario are not *distance-aware*, effectively rendering the non-parametric part of layer-specific SVGP obsolete, thus resulting in a fully parametric DGP from the first hidden layer up to the output layer (for in-depth insight into why this pathology occurs we refer the reader to Figure 3.7 and accompanying text). The second scenario for DGP will investigate the case where the non-parametric uncertainty for an OOD data point does get correctly propagated throughout the layers, thus making use of the non-parametric part of SVGP in each layer of the hierarchy. We lay out the necessary conditions in the following proposition. Before we proceed, we remind ourselves that a SVGP can be decomposed as $F(\cdot) \sim \underbrace{K_{fu}K_{uu}^{-1}U}_{G(\cdot)} + \underbrace{\mathcal{N}\left(0, K_{ff\cdot u}\right)}_{H(\cdot)}$.

Proposition 3.3.2. *Under the assumption that for any OOD point in input space x_{out} we have that $G_l(x_{out}) = K_{fu}K_{uu}^{-1}U_l \approx 0$ at each layer in a DGP, we can instead focus solely on the non-parametric components of DGP as a composition of $p(H_L) = p(H_L \mid H_{L-1}) \circ ... \circ p(H_1 \mid H_0)$ with $p\left(H_l \mid H_{l-1}\right) = \mathcal{N}\left(H_l|0, K_{ff\cdot u}\right)$ and $H_0 = X$. We consider two inducing points in the first hidden layer $\{Z_{1,1} = -c, Z_{1,2} = c\}$ with $c \geq 0$ and an OOD data point x_{out} in input space such that $p(H_1(x_{out})) \sim \mathcal{N}\left(0, \sigma_1^2\right)$. The approximate variance of the non-parametric component pertaining to the second layer will satisfy $v\left(H_2(x_{out})\right) \geq K_{ff\cdot u}\Big|_{F_1=0}$ if the following holds $l_2^2 \geq 2c^2$, where the Schur complement is evaluated at $F_1 = 0$.*

A proof of this proposition can be found in Appendix A.2.3.

Remark 3.3.2. *Reliable propagation of non-parametric uncertainty for OOD data points in DGP can only occur if the parametric posterior at each layer $\mathcal{N}\left(G_l \mid K_{fu}K_{uu}^{-1}m_{U_l}, K_{fu}K_{uu}^{-1}S_{U_l}K_{uu}^{-1}K_{uf}\right)$ manages to reliably map in-distribution points stemming from the previous layer in the intervals $I_{in-d}^1 = [-\infty, -3\sigma_{l-1}]$, respectively $I_{in-d}^2 = [3\sigma_{l-1}, \infty]$. This scenario can occur for example in a simple binary classification task, where the parametric posteriors at each layer have learned to reliably separate the two classes in hidden layer features. Considering that OOD data points from the previous layer have a predictive distribution $\mathcal{N}\left(0, \sigma_{l-1}^2\right)$, upon sampling 99.7% of samples from*

this distribution will be in the interval $I_{OOD} = [-3\sigma_{l-1}, 3\sigma_{l-1}]$. *Hence, under this assumption the intervals for OOD points and in-distribution points will be disjoint. Under an assumption of well-optimized inducing points' locations,* Z_{l-1} *will invariably find themselves in either* I^1_{in-d} *or* I^2_{in-d}, *with no locations in* I_{OOD}. *Using notation from Proposition 3.3.2,* $c = 3\sigma_{l-1}$ *and the approximate variance* $v\left(0, \sigma^2_{l-1}\right) \geq K_{ff \cdot u}$ *where the Schur complement is evaluated at* $F_{l-1} = 0$ *only if* $l_l^2 \geq 2c^2$.

To summarize our results so far, in this section we have shown the necessary conditions for DGP to maintain a comparatively higher total uncertainty for OOD points in comparison to in-distribution points. Through empirical examples, we have seen that the non-parametric variance collapses to zero even in toy regression datasets, thus making the DGP rely solely on its parametric components. Next, we have proposed a scenario in which the posterior DGP can reliably propagate non-parametric variance for OOD points. However, this latter scenario is not guaranteed to occur in practice. In the next section, we propose a variant of DGP which are guaranteed to propagate non-parametric variance for OOD points.

3.4 Distributional Deep Gaussian Processes

In this section we commence with desiderata for kernels in DGP, subsequently proposing the hybrid kernel (*i.e.*, kernel operationg on Euclidean $\&$ Wasserstein space) as a solution, alongside introducing the generative process and inference framework for our newly introduced model. Moreover, we derive the theoretical requirements of correctly propagating outliers through the hierarchy of this probabilistic formulation and show that it's easier satisfied in comparison to DGP.

3.4.1 Desiderata for kernels in DGP

I Maintenance of outlier status. While there are many definitions of what an outlier constitutes, for the purposes of this work we follow the most basic one introdued in Ruff et al. (2021) : *"An anomaly or outlier is an observation that deviates considerably from some concept of normality."*. The authors formalise it as follows: given data that resides in $X \in \mathbb{R}^D$, an outlier is a data point $x \in X$ that lies in a low probability region under $\mathcal{P}$, which is the *ground-truth law of normal behaviour* in a given task or application. such that the set of outliers is defined as $A = \{x \in X \mid p(x) \leq \xi\}$, with $\xi \geq 0$ being a threshold under which we consider data points to deviate sufficiently from what normality constitutes. GP constitute an easy plug-in method to detect if points are inside the data-manifold or not. Distributional uncertainty as defined in equation (2.53) can be used as proxy for OOD detection due to its *distance-awareness* properties (see section 1.2). Therefore we desire DGP with similar OOD detection properties as standard GP at every layer in the former's hierarchy.

II Smoothness in hidden layers. Imposing smoothness in neural networks' hidden layers has been linked to increased generalization, robustness to input perturbations or reliable uncertainty estimates (Rosca et al. 2020). With the latter property in mind, recent work (Amersfoort et al. 2021; Liu et al. 2020a) has proposed bi-Lipschitz constraints for extracting features to be fed into a GP. This paradigm is based on the notion that objects which were close in previous layers should also be close going forward in the hierarchy, thus avoiding *feature-collapse* (Amersfoort et al. 2021). Consequently, kernels in DGP should provide a certain guarantee of smoothness alongside guarding against *feature-collapse*.

3.4.2 Hybrid kernel

The term *Distributional Gaussian Process* was first introduced in Bachoc et al. (2017) to describe a shallow GP that operates on measures using a Wasserstein-2 based kernel (see equation (3.2)) and for purposes of differentiating it from subsequent models it will be coined as *DistGP-Bahoc* throughout the remainder of the chapter. We first investigate prior function space properties if the latter formulation is extended to the multi-layered case and then introduce our proposed kernel and generative process.

Considering the case of Euclidean input data X, the first layer is governed by a GP $p(F_1) = \mathcal{N}(0, K_{F_0F_0})$, where for notation purposes we considered $F_0 = X$. We immediately see that the prior has the capability to capture correlations between data points. Adding a new layer given by *DistGP-Bahoc* operating directly on $p(F_1)$, where we only take diagonal terms of its covariance matrix to compute the kernel, results in $K^{W_2}_{F_{1,i},F_{1,j}} = \sigma_2^2 \exp \frac{-W_2^2(\mathcal{N}(0,\sigma_1^2),\mathcal{N}(0,\sigma_1^2))}{l_2^2} = \sigma_2^2$ for all data entries. Therefore, correlations between different data points are ignored as we arrive at the prior on $p(F_2)$:

$$p(F_2) = \mathcal{N}\left(\begin{bmatrix}0\\ \vdots \\ 0\end{bmatrix}, \begin{bmatrix}\sigma_2^2 & \dots & \sigma_2^2 \\ \vdots & \ddots & \vdots \\ \sigma_2^2 & \dots & \sigma_2^2\end{bmatrix}\right) . \tag{3.4}$$

Consequently, this type of construction will exhibit an overly-correlated prior which will cause samples to collapse to set of finite values, thereby not satisfying Desiderata II. The introduction of PCA mean function (see section 2.4.5) to the generative process can solve this issue (see section 4.2). However, we desire a generative process that is without such pathologies in the zero-mean function case.

Sampling hidden layers' features in DGP is of utmost importance towards introducing correlations between data points. On the other hand, the Wasserstein-2 kernel (equation (3.2)) directly applied on hidden layers' resulting multivariate Gaussians is attractive as it takes into account differences between variance terms. The hindsight is that differences between low and high variance terms in the multivariate normals would enable a finer separation of data points which were OOD compared

to in-distribution as detected by the previous layer. With these observations in mind, we argue that both stochastic (sampling hidden layer features in DSVI framework for DGP) and deterministic versions (Wasserstein-2 kernels directly applied on hidden layer GP) are important from entirely complementary viewpoints. We seek to combine these two formulations in a single one that preservers both key properties, we can now write the generative process of this new probabilistic framework coined Distributional Deep Gaussian Processes (DDGP) for 2 layers:

$$F_1 \sim \mathcal{N}(0, K_{ff}) \tag{3.5}$$

$$F_1^{sth} = m(F_1) + \sqrt{v(F_1)}\epsilon,\ \epsilon \sim \mathcal{N}(0, I_n) \tag{3.6}$$

$$F_1^{det} = \mathcal{N}(m(F_1), \mathrm{diag}\,[v(F_1)]) \tag{3.7}$$

$$F_2 \sim \mathcal{N}\left(0, k_{hybrid}\left(\{F_1^{sth}, F_1^{det}\}, \{F_1^{sth}, F_1^{det}\}\right)\right)\ . \tag{3.8}$$

Intuitively, this generative process implies keeping track of a *stochastic* and a *deterministic* component of the same SVGP at any given hidden layer. In Figure 3.3 we provide a schematic of the differences in the generative process of DGP and DDGP.

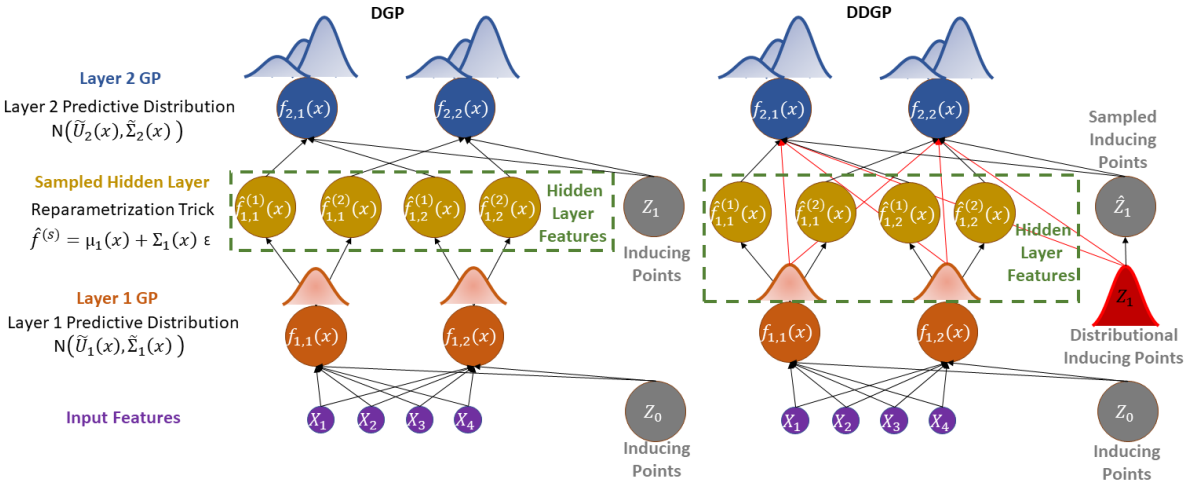


Figure 3.3: Conceptual differences between DGP and DDGP. Fundamentally, DGP operate in Euclidean space of sampled hidden layer features, while DDGP also operate on the Wasserstein space of underlying hidden layer GP. **Layer 1:** There is no difference between DGP and DDGP. **Layer 2:** Distributional inducing points consist of a Gaussian distribution, for which we have the first two moments as model parameters. Hybrid kernel use both the feature space samples and the underlying Gaussian distribution from which the samples originated. To compute terms such as K_{uu} or K_{uf} one needs to sample also from distributional inducing points; Red objects in the diagram signify differences between DDGP compared to DGP.

We took inspiration from Varifold theory (see subsection 3.2.2 for details) and devised a kernel operating on both Euclidean and Wasserstein-2 space, thereby incorporating both stochastic and deterministic pathways. Considering again two data points in the prior space of F_1, we have $p\begin{pmatrix} F_{1,i} \\ F_{1,j} \end{pmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & K_{i,j}^{SE} \\ K_{j,i}^{SE} & \sigma_1^2 \end{bmatrix}\right)$ and their associated samples $\{f_i, f_j\}$. To compute

Wasserstein-2 distances using the previous layer's multivariate Normal, we take the diagonal of the covariance matrix and we denote $\mu_i = \mathcal{N}(0, \sigma_1^2)$; $\mu_j = \mathcal{N}(0, \sigma_1^2)$. We can now introduce the hybrid kernel:

$$k^H\left(\{\mu_i, f_i\}, \{\mu_k, f_j\}\right) = k^{SE}(f_i, f_j) \exp \sum_{d=1}^{D} \frac{-W_2^2(\mu_{i,d}, \mu_{j,d})}{l_d^2} \,. \tag{3.9}$$

Prior samples of F_2 will be correlated due to the Euclidean part of this kernel, whereas the Wasserstein-2 component will always be equal to 1, hence neither introducing or removing extra correlations between data points. We shall soon see that this results in the prior of DDGP equating that of DGP. To make this clear, we can explicitly calculate it:

$$p\begin{pmatrix} F_{2,i} \\ F_{2,j} \end{pmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_2^2 \exp -\frac{-W_2^2(\mu_i,\mu_i)}{l^2} & K_{i,j}^{SE} \exp -\frac{-W_2^2(\mu_i,\mu_j)}{l^2} \\ K_{j,i}^{SE} \exp -\frac{-W_2^2(\mu_j,\mu_i)}{l^2} & \sigma_2^2 \exp -\frac{-W_2^2(\mu_j,\mu_j)}{l^2} \end{bmatrix}\right) \tag{3.10}$$

$$\sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_2^2 & K_{i,j}^{SE} \\ K_{j,i}^{SE} & \sigma_2^2 \end{bmatrix}\right) \,, \tag{3.11}$$

since the Wasserstein-2 distance between two equal distributions (*i.e.*, μ_i and μ_j) is zero.

Remark 3.4.1. *In prior space, a hierarchical construction based on hybrid kernels (see equation* (3.9)*) for hidden layers is equivalent to a DGP. In posterior space, this no longer holds true with several different properties which we will subsequently outline.*

Connections between Varifold kernel and Hybrid Kernel. Whereas usually during training we only sample once from $q(F_l)$ at each layer in order to obtain the samples that get used in the Euclidean part of the kernel, one could sample arbitrary times, thereby obtaining a collection of Gaussian distributions. In this case, we can re-express our hybrid kernel in the multi-sample scenario:

$$\sum_{s=1}^{S}\sum_{s'=1}^{S} k^{SE}(f_i^{(s)}, f_j^{(s')}) \exp \sum_{d=1}^{D} \frac{-W_2^2(\mu_{i,d}^{(s)}, \mu_{j,d}^{(s')})}{l_d^2} \,. \tag{3.12}$$

We can observe similarities with the varifold kernel in detailed in A.1, where we compute a distance between submanifolds, taking into consideration Euclidean distances between discrete points on the manifolds' surface, combined with information pertaining to the points' geometry. For hybrid kernels as applied to DDGP, we compare two sets of multivariate Gaussians, taking into account Euclidean differences between points sampled from the underlying distributions, but also considering geometric differences between these distributions as captures by the Wasserstein-2 distance. A visual depiction of these different but similar objectives is provided in figure 3.1.

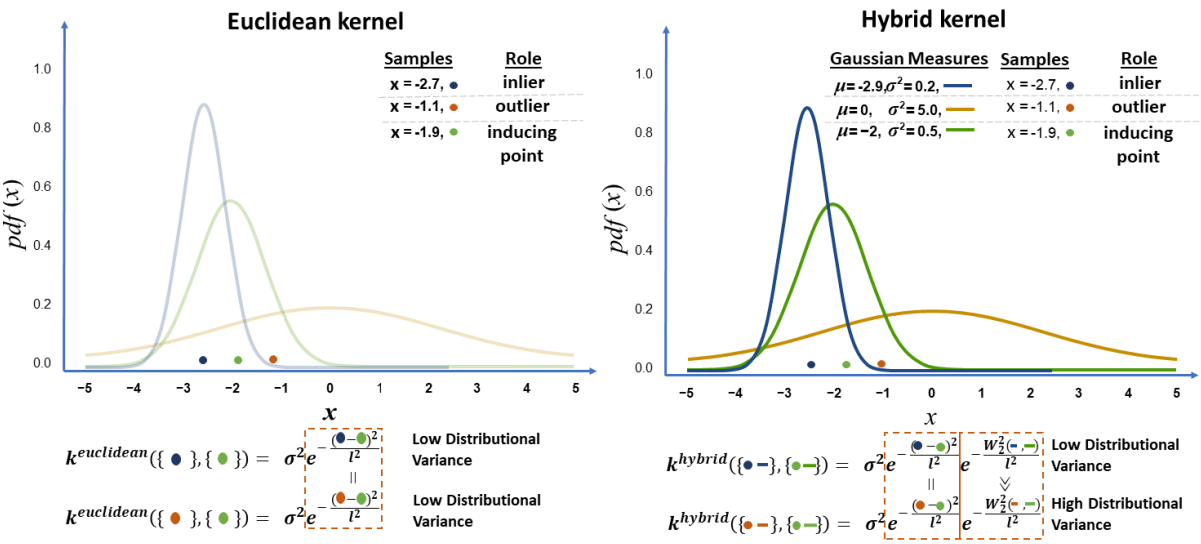


Figure 3.4: Conceptual difference between euclidean and hybrid kernel. Explicit difference between moments in Wasserstein-2 distance ensures *distance-awareness* for DDGP. Differences between samples in Euclidean space are similar for in-distribution and outlier points with respect to inducing point location. The difference between the predictive variance of the outlier points with respect to the distributional inducing points' variance is higher than that of the in-distribution point. This ensures higher distributional uncertainty for the former.

Hybrid kernels and OOD detection. The key motivation behind the introduction of hybrid kernels can be succinctly described by a simple toy example. Considering two-layered DGP and DDGP, we assume an in-distribution point x_{in-d} to have low total variance in hidden layer F_1 and an OOD point x_{ood} to have high total variance. In the DGP case, upon sampling from $q(F_1(x_{in-d}))$ and $q(F_1(x_{ood}))$ we can end up with samples which are equally distant with respect to inducing points' location Z_1. If this occurs, then non-parametric variance will be equal for x_{in-d} and x_{ood} in F_2. Hence, what was initially flagged as OOD in the first hidden layer will be considered as in-distribution by the second hidden layer, thus not satisfying Desiderata I. Conversely, in the DDGP case and under the assumption that the variance of distributional inducing points' locations Z_2 is almost equal in distribution to the total variance of in-distribution points in F_1, the Wasserstein-2 component of the hybrid kernel will notice that there is a higher distance between the now distributional inducing point location and x_{ood}, as opposed to the distance of the former with x_{in-d}. Then, the non-parametric variance (*i.e.*, distributional uncertainty) of x_{ood} will be higher than that of x_{in-d}, thus satisfying Desideratum I. A visual depiction of this case study is illustrated in Figure 3.4.

3.4.3 Evidence lower bound

In this subsection we outline the generative process of constructing DDGP.

First, we remind ourselves that a DGP is defined as a stack of shallow GP acting as the prior:

$$\underbrace{p(Y|F_L)}_{\text{likelihood}} \underbrace{\prod_{l=1}^{L} p(F_l|F_{l-1}, U_l; Z_{l-1})p(U_l)}_{\text{prior}} , \tag{3.13}$$

where for brevity of notation we denote $F_0 = X$. The SVGP between hidden layers are treated as being noise-free. As the prior is analytically intractable to integrate, Salimbeni and Deisenroth (2017) have suggested to sample from each hidden layer of the DGP in order to obtain unbiased stochastic gradients. The difference between DGP and DDGP resides in the fact that for $l \geq 2$ we not only sample from the marginal of the data points $q(F_l; F_{l-1})$, but we also explicitly use this marginal in the computation of the next layer's kernel. More concretely, we have the following joint density prior $p(Y, \{F_l, U_l\}_{l=1}^L)$:

$$\underbrace{p(Y|F_L)}_{likelihood} \underbrace{\prod_{l=2}^{L} p(F_l|F_{l-1}, U_l; Z_{l-1})p(U_l)}_{\text{Wasserstein \& Euclidean space}} \underbrace{p(F_1|U_1; X)p(U_1)}_{\text{Euclidean space}} . \tag{3.14}$$

We introduce a factorized posterior between layers and dimensions of the following form : $q(\{F_l\}_{l=1}^L, \{U_l\}_{l=1}^L) = p(F_L|U_L; Z_{L-1}) \prod_{l=1}^L q(U_l)$, where $q(U_l)$ is taken to be a multivariate Gaussian with mean m_{U_l} and variance S_{U_l}. The posterior over $q(F_1)$ is similar to the DGP case, with the difference residing in $l \geq 2$, where we have the following conditional DistGP:

$$p(F_l|U_l; Z_{l-1}, F_{l-1}) = \mathcal{N}(F_l \mid K_{fu}^H {K_{uu}^H}^{-1} U_l, K_{ff}^H - Q_{ff}^H) . \tag{3.15}$$

After sampling from $q(F_{l-1})$ and $Z_{l-1} \sim \mathcal{N}\left(Z_{l-1} \mid M_{Z_{l-1}}, \Sigma_{Z_{l-1}}\right)$, we proceed to integrate out U_l, arriving at:

$$q(F_l) = \mathcal{N}(F_l \mid K_{fu}^H {K_{uu}^H}^{-1} m_{U_l}, K_{ff}^H - Q_{ff}^H + K_{fu}^H {K_{uu}^H}^{-1} S_{U_l} {K_{uu}^H}^{-1} K_{uf}^H) , \tag{3.16}$$

where in the computation of the hybrid kernels we utilise $q(F_{l-1})$ (*i.e.*, its first 2 moments) for computing the Wasserstein-2 distance, respectively samples from $q(F_{l-1})$ in the Euclidean part. It is worth highlighting the fact that for $l \geq 2$, Z_l follow multivariate Normal distributions and are called distributional inducing points (see Figure 3.3 for illustration). Further derivations similar to the ones done for DGP in subsection 2.4.3 will lead us to the following ELBO:

$$\mathcal{L}_{DDGP} = \mathbb{E}_{\{q(F_l, U_l\}_{l=1}^L} p(Y|F_L) - \sum_{l=1}^{L} \mathbb{E}_{Z_l} \left[\mathrm{KL}\left[q(U_l) \| p(U_l)\right]\right] . \tag{3.17}$$

Having now defined the generative process and the training procedure for DDGP, we can focus on seeing if our proposed model satisfies the desiderata laid out at the beginning of the section.

3.4.4 Propagation of variance in DDGP

In this subsection, we investigate what are the necessary requirements for reliable propagation of non-parametric variance through the hierarchy for OOD data points.

Proposition 3.4.1. *We consider the approximate posterior DDGP as a composition of functions* $q_L = q_L \circ ... \circ q_1$ *with* q_l *being given by* $q(F_l|F_{l-1}) = \int p(F_l|U_l, F_{l-1})q(U_l)\,\mathrm{d}U_l = \mathcal{N}(F_l \mid \tilde{U}_l(F_{l-1}), \tilde{\Sigma}_l(F_{l-1}))$. *We consider an OOD data point* x_{out} *in input space such that* $\tilde{\Sigma}_1(x_{out}) = \sigma_1^2$ *and* $\tilde{U}_1(x_{out}) = 0$, *respectively* x_{in} *to be in-distribution, with* $\tilde{\Sigma}_1(x_{in-d}) = V_{in-d} \leq \sigma_1^2$ *and* $\tilde{U}_1(x_{in-d}) = M_{in-d}$. *We assume that* M_{Z_1} *are equidistantly placed between* $[-3\sigma_1, 3\sigma_1]$ *and* $\Sigma_{Z_1} \leq \sigma_1^2$. *The approximate variance in the second layer of* x_{out} *will be higher than* x_{in-d} *if the following hold:*

$$\sigma_1^2 \gg \Sigma_{Z_2} \text{ and } V_{in-d} \approx \Sigma_{Z_2}$$

A proof of this proposition can be found in Appendix A.2.4.

Remark 3.4.2. *In opposition to DGP which require a specific pattern of first order derivatives surrounding 0, DDGP rely less on this constraint as the above inequality is already providing a significant difference in non-parametric variance between outlier and inlier points, thus satisfying Desideratum I. For any layer* l, Σ_{Z_l} *will converge towards values of the parametric posterior variance of in-distribution points, since the inducing points are encouraged to gravitate towards the in-distribution manifold at layer* l. *In well optimized scenarios, the parametric posterior variance will be reduced compared to the prior variance (*σ_{l-1}^2*). Considering these observations, the conditions laid out in Proposition 3.4.1 will be satisfied in practice.*

3.4.5 *Feature-collapse* in (D)DGP

Dunlop et al. (2018) provide an in-depth analysis of function space properties DGP have, which DDGP are equivalent to in prior space (see subsection 3.4.2). We are interested to find what are the necessary requirements, given a few constraints, of the function space values to collapse in the posterior for in-distribution data points.

We now introduce some notation conventions. For $D \in \mathcal{R}^d$ and $\{D_l \in \mathcal{R}^{d_l}\}_{l=1}^L$, where d_l is the number of dimensions in the l-th layer of the hierarchy. We consider the functions $f_l : D_{l-1} \to \mathcal{R}^{d_l}$, with a DGP being defined as composition of functions: $f_n\left(\cdots f_2\left(f_1\left(x\right)\right)\right)$.

For a DGP at layer n we assume the following posterior conditional formulation for two data points $x,\ x^* \in D_{in-d}$, respectively:

$$p\begin{pmatrix} f_n(x) \\ f_n(x^*) \end{pmatrix} = \mathcal{N}\left(\begin{bmatrix} K_{xu}K_{uu}^{-1}U_n \\ K_{x^*u}K_{uu}^{-1}U_n \end{bmatrix}, \begin{bmatrix} K_{xx} - Q_{xx} & K_{xx^*} - Q_{xx^*} \\ K_{x^*x} - Q_{x^*x} & K_{x^*x^*} - Q_{x^*x^*} \end{bmatrix}\right) \tag{3.18}$$

For subsequent derivations, we denote $\mu(x) = \mathcal{N}\left(K_{xu}K_{uu}^{-1}U_n, K_{xx} - Q_{xx}\right)$.

Proposition 3.4.2. *We assume f_0 to be bounded on bounded sets almost-surely. For a DGP, if at each layer l we have satisfied the following inequality $\frac{\sigma_l^4}{4l_l^4}\|K_{uu}^{-1}U_l\|_2^2\left(f_l(x) - f_l(x^*)\right)^2 < 1$, alongside $Q_{ff} \approx K_{ff}$, given two in-distributions points x and x^* we have the following result:*

$$P\left(\|f_n(x) - f_n(x^*)\|_2 \to 0 \mid \{U_l\}_{l=1}^n\right) = 1$$

A proof of this proposition can be found in Appendix A.2.5.

Remark 3.4.3. *From the constraint $\frac{\sigma_l^4}{4l_l^4}\|K_{uu}^{-1}U_l\|_2^2\left(f_l(x) - f_l(x^*)\right)^2 < 1$ to ensure* feature-collapse*, we can see that in the converse case, of avoiding* feature-collapse*, this can be assured by either increasing the kernel variance σ_l^2 or $\|K_{uu}^{-1}U_l\|_2^2$, which are both responsible for the amplitude of sampled functions. Moreover, another pathway to avoid* feature-collapse *is to increase the lengthscale l_l^2. Intuitively, this will ensure that unseen test points close to an inducing point will reliably have their predicted mean almost equal to that respective inducing point's value. Conversely, for lower lengthscale values, the predicted mean for an unseen points will be a weighted version of inducing points' output values across a larger distance, hence favouring similar values for unseen points, which will eventually lead to* feature-collapse*.*

Now we are interested in deriving similar constraints for DDGP to suffer from *feature-collapse* and possibly seeing if there are any differences.

Proposition 3.4.3. *We assume f_0 to be bounded on bounded sets almost-surely. For a DDGP, if at each layer l except the first one we have satisfied the following inequality $\frac{\sigma_l^4}{4l_l^4}\|K_{uu}^{-1}U_l\|_2^2[f_l(x) - f_l(x*) + \frac{\|m(f_l(x)) - m(f_l(x^*))\|_2^2 + \|v(f_l(x)) - v(f_l(x^*))\|_2^2}{f_l(x) - f_l(x*)}]^2 \leq 1$, given two in-distributions points x and x^* s.t. $Q_{ff} \approx K_{ff}$ we have the following result:*

$$P\left(\|f_n(x) - f_n(x^*)\|_2 \to 0 \mid \{U_l\}_{l=1}^n\right) = 1$$

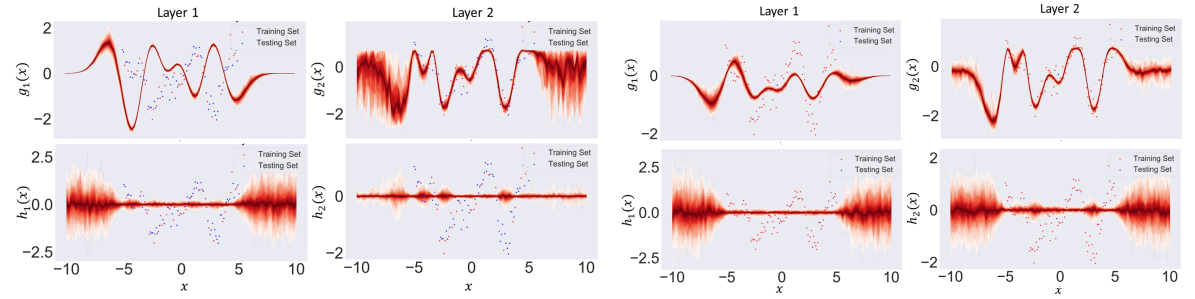
A proof of this proposition can be found in Appendix A.2.6.

Remark 3.4.4. *The constraint for DDGP to suffer from* feature-collapse *$\frac{\sigma_l^4}{4l_l^4}\|K_{uu}^{-1}U_l\|_2^2[f_l(x) - f_l(x*) + \frac{\|m(f_l(x)) - m(f_l(x^*))\|_2^2 + \|v(f_l(x)) - v(f_l(x^*))\|_2^2}{f_l(x) - f_l(x*)}]^2 \leq 1$, given two in-distributions points x and x^* s.t. $Q_{ff} \approx K_{ff}$ is similar to the one for DGP expect the final term in the brackets. We can see that for any given value of $f_l(x) - f_l(x^*)$, since $\|m(f_l(x)) - m(f_l(x^*))\|_2^2 + \|v(q(f_l(x)) - v(f_l(x^*))\|_2^2$ is always positive, it will result in $[f_l(x) - f_l(x*) + \frac{\|m(f_l(x)) - m(f_l(x^*))\|_2^2 + \|v(f_l(x)) - v(f_l(x^*))\|_2^2}{f_l(x) - f_l(x*)}]^2 \geq [f_l(x) - f_l(x*)]^2$. This means that DDGP are more suitable to guard against* feature-collapse *than DGP ceteris paribus. Moreover, we posit that the dependence on moment differences between two data points will embed DDGP with* smoother *properties in comparison to DGP, since the moments are computed using squared exponential kernels, which will produce smoother functions. With these*

two properties in mind, we can affirm that DDGP satisfy Desideratum II.

3.5 Experiments

Collapse of non-parametric variance in DGP. Duvenaud et al. (2014) highlighted a certain pathology when stacking GP, respectively the increasingly more non-injective mappings as the network increases in number of layers. In this section we highlight another pathology of DGP with zero-mean functions, namely the collapse of non-parametric uncertainty. In Figure 3.2 we highlighted this phenomenon for a 3-layer DGP architecture, however this is also valid for networks with even just two hidden layers (Figure 3.5). We can observe that this pathology does not constitute a problem for DDGP due to the introduction of hybrid kernels. Moreover, we empirically find that as we increase the number of inducing points, this erroneous behaviour is accentuated even in relatively small networks with just two hidden layers (Figure 3.6).



(a) DGP (zero-mean function) **(b)** DDGP (zero-mean function)

Figure 3.5: Non-parametric variance is correctly propagated in DDGP. Top row: Layer-wise parametric component $G_l(x)$ showing epistemic uncertainty; **Bottom row:** Layer-wise non-parametric component $H_l(x)$ showing distributional uncertainty; We can observe that for the hidden layers of DGP it collapses to almost zero, thus rendering $H_l(x)$ almost useless except for the first layer.

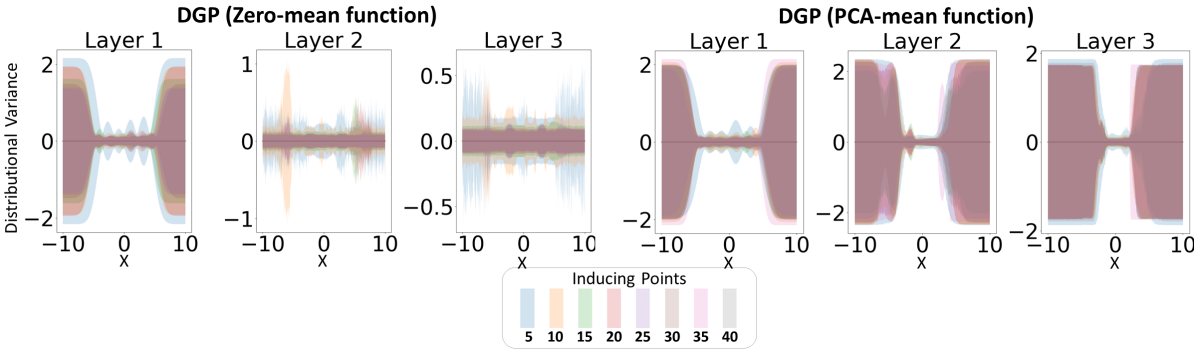


Figure 3.6: Progressive collapse of distributional uncertainty in DGP. Distributional uncertainty of DGP trained on a toy dataset with data between -5.0 and 5.0 with a varying number of inducing points. X axis denotes input space.

To understand what is causing this pathology, we take a simple case study of a DGP with two hidden layers trained on a toy regression dataset (Figure 3.7). Taking a clear outlier in input space,

say the data point situated at -7.5, it gets correctly identified as an outlier in the mapping from input space to hidden layer space as given by its distributional variance. However, its outlier property dissipates in the next layer after sampling, as it gets mapped to regions where the next GP assigns inducing point locations. This is due to points inside the data manifold getting confidently mapped between -2.5 and 2.5 in hidden layer space. Consequently, what was initially correctly identified as an outlier will now have its final distributional uncertainty close to zero. Adding further layers, will only compound this pathology.

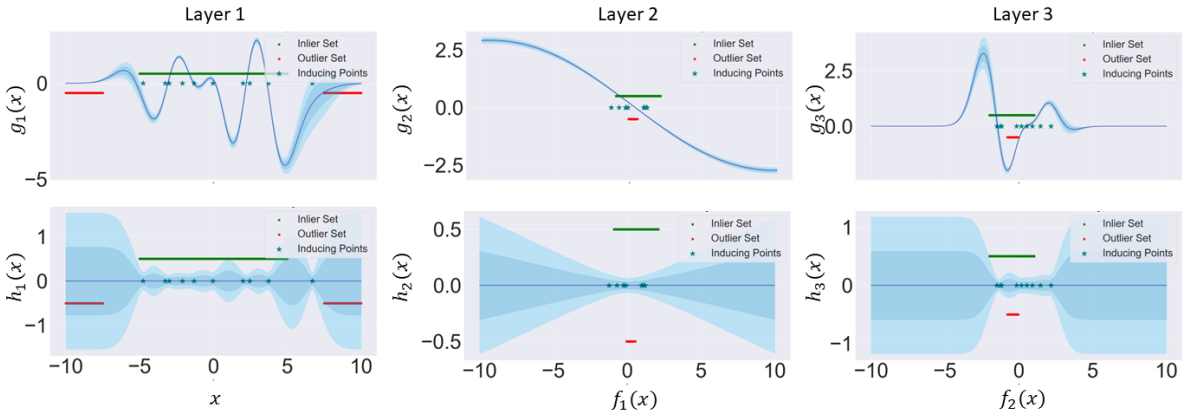
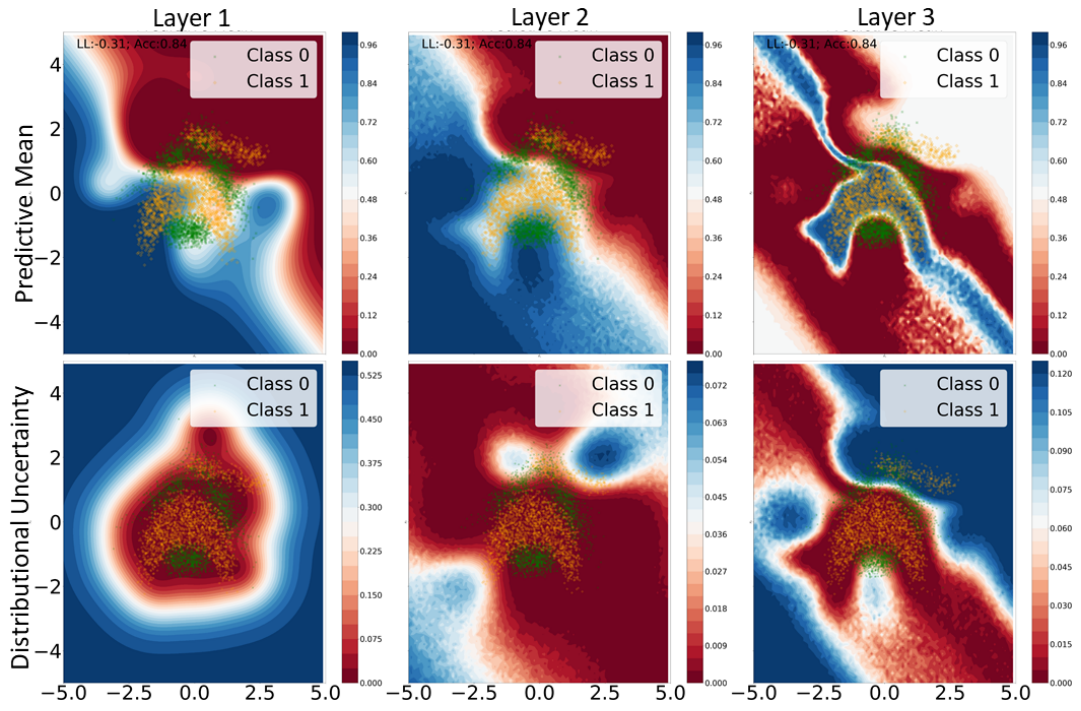
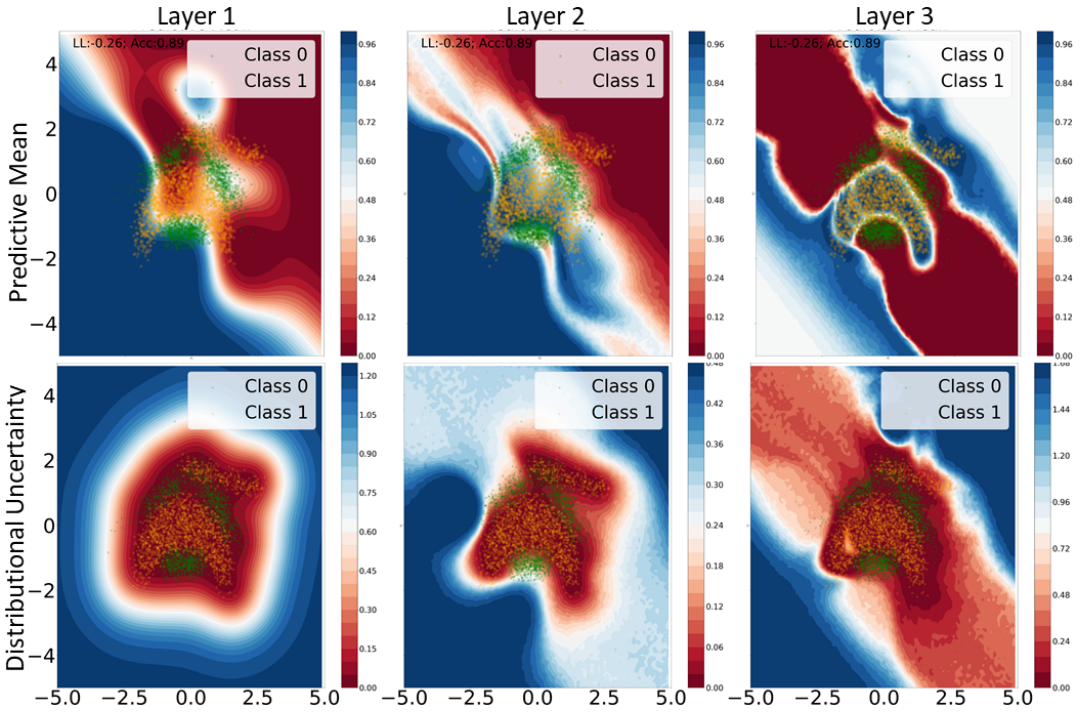


Figure 3.7: Rationale behind collapse of non-parametric uncertainty. Layer-wise decomposition of uncertainty into parametric and non-parametric for a zero-mean DGP. OOD points in input space (red dots) get sampled in hidden layers close to in-distribution points (green dots), hence close to inducing point locations.

Interplay between PCA mean functions and non-parametric uncertainty. We now investigate the effect that PCA mean function has on distributional uncertainty. In the case that the hidden layers' size is equal or larger than input size, then DGP will retain their OOD capabilities (Figure 3.6). Conversely, if the hidden layers' size is considerably lower than input size (such as image classification) we posit that it will negatively affect OOD capabilities. We use the "banana" dataset and train our proposed models with 2 hidden layers and 10 inducing points. The DGP distributional uncertainty is guided by the direction of the PCA projection of the data, thereby widely extrapolating low uncertainty regions to spatial locations where there is no data. For DDGP the PCA projection only slightly influences distributional variance in the last layer (Figure 3.8).



(a) DGP (PCA mean function)



(b) DDGP (PCA mean function)

Figure 3.8: PCA mean functions influence distributional uncertainty in DGP. Top row: Layer-wise predictive mean showcasing how the decision boundaries are formed; **Bottom row:** Layer-wise distributional uncertainty, with values closer to dark red indicating low values, which translate to increased confidence in predictions; DGP (a) and DDGP (b) models trained on the "banana" dataset. Distributional uncertainty in hidden layers is lower on the data manifold compared to outside the manifold for DDGP. Conversely, DGP widely extrapolate low distributional uncertainty outside the data manifold.

Smoothness properties From Proposition 3.4.2 and 3.4.3 we would expect that DDGP exhibit smoother hidden layers features. Using our previous experiment on the "banana" dataset, we keep track of correlations between two focus points with respect to full input space (Figure 3.9). The increased "locality" of features in layer 3 is striking in the case of DDGP. Moreover, we notice even in layer 2 a slight difference in smoothness between the two models.

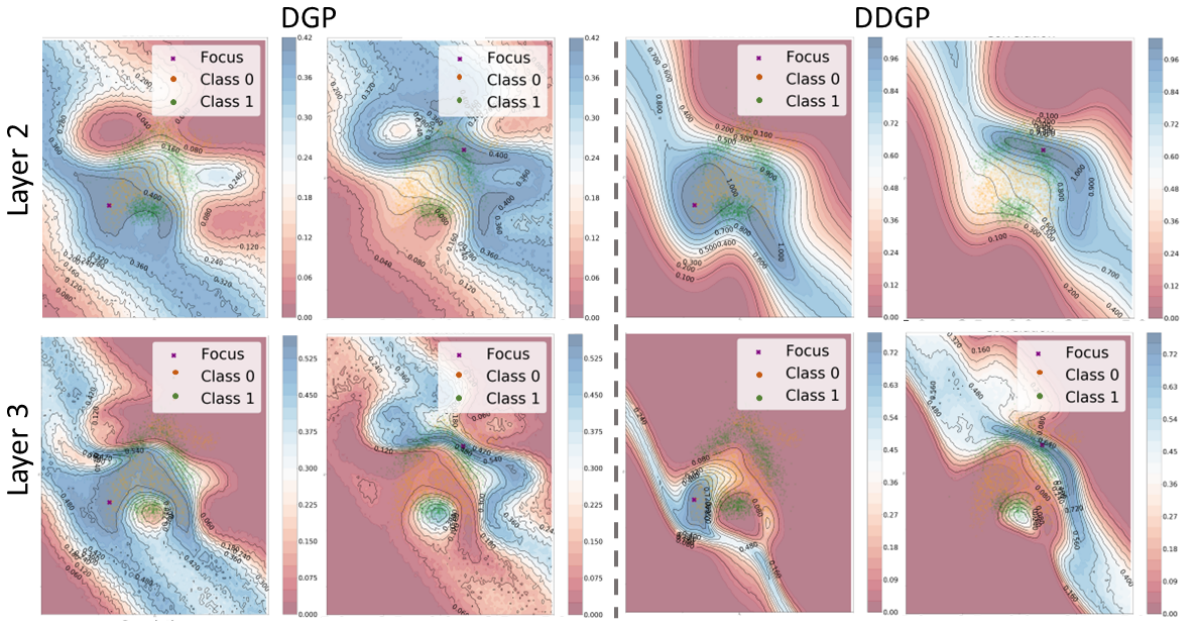
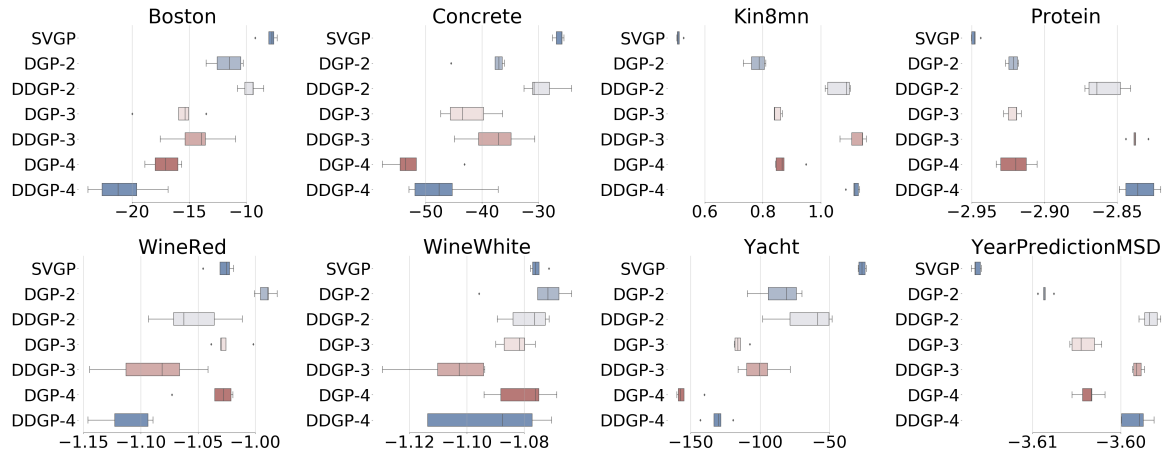


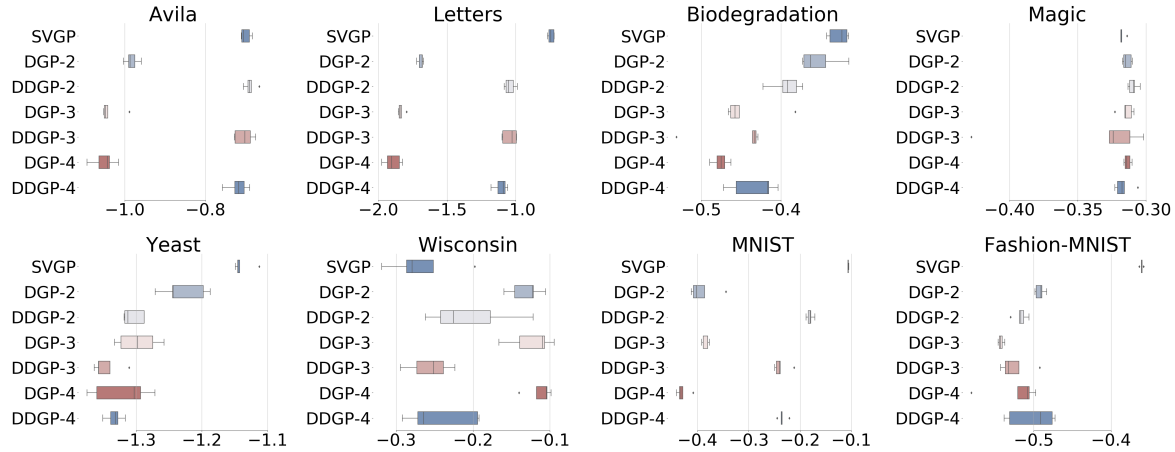
Figure 3.9: Smoothness comparison between DGP and DDGP. Correlation of two focus data points (purple dots) with respect to full input space spectrum at hidden layers for DGP (PCA mean function) and DDGP (PCA mean function) trained on "banana" dataset. DDGP (right panel) exhibit smoother and more "local" correlations.

Performance on real-life data. Besides testing whether DDGP surpass DGP in terms of OOD detection, it is also of vital importance to see whether DDGP manage to obtain calibrated predictive distributions on medium and large scale datasets. For this we evaluate our models on datasets from the UCI machine learning dataset repository. All of the experiments were run with the same initializations of variational parameters and kernel hyperparameters, with the goal of comparing DDGP with DGP for 2, 3 and 4 layers. We used 100 inducing points at each layer, with 5 dimensions per hidden layer and PCA mean functions. The baseline SVGP model also uses 100 inducing points. Further details regarding training and initialization are given in Appendix A.3.

Regression results. In terms of predictive log-likelihood values on test data, DDGP generally surpass their DGP counterparts on regression datasets, with an overall diminished performance seen only on the "WineRed" and "WineWhite" datasets (see Figure 3.10a for a visual inspection). From Table 3.1a we can notice that the best performing model across datasets in terms of rank is the DDGP with 2 layers. DDGP models outperform their DGP counterpart in terms of average rank regardless of the number of layers.



(a) Test log-likelihood regression datasets



(b) Test log-likelihood classification datasets

Figure 3.10: Results on UCI datasets. Top: Test log-likelihood values on regression datasets. **Bottom:** Test log-likelihood values on classification datasets. **All subplots**: 10 different splits of the dataset for each model are taken into the composition of each boxplot. Higher values (to the right) indicate better model fit.

Classification results. In terms of predictive testing set results on classification datasets, the behaviour is roughly similar, with noticeable improved performance of DDGP on the following datasets: "Avila", "Letters" and "MNIST" (see Figure 3.10b for a visual inspection). Conversely, the same can be said for DGP on the "Wisconsin" dataset. Just like in the case of regressions tasks, DDGP models outperform their DGP counterparts regardless of number of layers. We can also notice that SVGP is the best performing model across datasets in terms of average rank (see Table 3.1b).

Out-of-distribution detection. We adapt DDGP to the architecture introduced in Deep Convolutional GP (DCGP) (Blomqvist et al. 2018), obtaining the Distributional Deep Convolutional GP (DistDCGP). The goal is to investigate their OOD detection capabilities on real data. To detect outliers we need a scalar measure encompassing the uncertainty present across all classes. For this we use

Dataset	SVGP	DGP-2	DDGP-2	DGP-3	DDGP-3	DGP-4	DDGP-4
Boston	**-7.97 (0.66)**	-11.65 (1.23)	-9.62 (0.77)	-15.95 (2.16)	-14.26 (2.16)	-17.09 (1.20)	-20.81 (2.45)
Concrete	**-26.25 (0.75)**	-38.50 (3.54)	-29.27 (2.98)	-42.47 (3.98)	-37.58 (4.87)	-52.09 (4.90)	-46.96 (5.65)
Kin8mn	0.51 (0.00)	0.77 (0.02)	1.06 (0.03)	0.84 (0.01)	**1.12 (0.03)**	0.87 (0.03)	1.11 (0.01)
Protein	-2.948 (0.002)	-2.921 (0.003)	-2.858 (0.012)	-2.921 (0.004)	-2.837 (0.004)	-2.920 (0.010)	**-2.834 (0.010)**
WineRed	-1.02 (0.00)	**-0.99 (0.00)**	-1.05 (0.02)	-1.02 (0.01)	-1.08 (0.03)	-1.03 (0.01)	-1.11 (0.02)
WineWhite	-1.075 (0.002)	**-1.074 (0.011)**	-1.078 (0.006)	-1.082 (0.004)	-1.106 (0.013)	-1.080 (0.009)	-1.092 (0.018)
Yacht	**-26.52 (2.31)**	-85.60 (14.38)	-66.75 (18.97)	-114.92 (4.05)	-99.85 (12.97)	-154.35 (7.24)	-130.86 (7.66)
YearPredictionMSD	-3.616 (0.000)	-3.608 (0.000)	**-3.596 (0.000)**	-3.604 (0.001)	-3.598 (0.000)	-3.603 (0.001)	-3.598 (0.001)
Average ranks	3.75 (2.64)	3.77 (1.82)	**2.72 (1.28)**	4.6 (1.01)	3.67 (1.91)	4.925 (1.53)	4.55 (2.36)

(a) Results on UCI regression datasets.

Dataset	SVGP	DGP-2	DDGP-2	DGP-3	DDGP-3	DGP-4	DDGP-4
Avila	-0.69 (0.01)	-0.98 (0.01)	**-0.68 (0.01)**	-1.03 (0.02)	-0.70 (0.02)	-1.05 (0.02)	-0.71 (0.02)
Letters	**-0.74 (0.01)**	-1.69 (0.01)	-1.04 (0.03)	-1.83 (0.02)	-1.04 (0.04)	-1.89 (0.05)	-1.10 (0.04)
Biodegradation	**-0.32 (0.01)**	-0.35 (0.02)	-0.39 (0.01)	-0.44 (0.03)	-0.45 (0.03)	-0.47 (0.00)	-0.43 (0.02)
Magic	-0.317 (0.001)	-0.313 (0.002)	**-0.309 (0.003)**	-0.314 (0.004)	-0.338 (0.045)	-0.313 (0.002)	-0.316 (0.005)
Yeast	**-1.13 (0.01)**	-1.22 (0.03)	-1.30 (0.01)	-1.29 (0.02)	-1.34 (0.01)	-1.32 (0.03)	-1.33 (0.01)
Wisconsin	-0.26 (0.04)	-0.13 (0.01)	-0.20 (0.05)	-0.12 (0.02)	-0.25 (0.02)	**-0.11 (0.01)**	-0.24 (0.04)
MNIST	**0.10 (0.00)**	-0.39 (0.02)	-0.18 (0.00)	-0.38 (0.00)	-0.23 (0.01)	-0.42 (0.01)	-0.23 (0.00)
Fashion-MNIST	**-0.36 (0.00)**	-0.49 (0.00)	-0.51 (0.00)	-0.54 (0.00)	-0.52 (0.01)	-0.52 (0.03)	-0.50 (0.02)
Average ranks	**2.35 (2.05)**	3.52 (1.62)	3.02 (1.40)	4.95 (1.75)	4.57 (1.73)	5.2 (2.01)	4.37 (1.51)

(b) Results on UCI classification datasets.

Table 3.1: Average test log-likelihood (standard error) results for ten splits on UCI regression datasets. Higher values indicate better model fit; Average rank (standard error) based on test log-likelihood results across all datasets and splits. Lower rank values indicate better position in hierarchy. Highlighted in bold for each row is the best result.

the distributional differential entropy introduced in equation (2.55). This has the effect to measure the overall distributional uncertainty in logit space, with higher values representing more uncertainty.

In these experiments we assess the capacity of our model to detect dataset shift by training it on MNIST (LeCun 1998) and comparing the uncertainty measures computed on the testing set of MNIST (*i.e.*, in-distribution data) with the ones obtained from NotMNIST (Bulatov 2011) and SVHN Netzer et al. (2011), which are out-of-distribution. Further information about these datasets alongside sample figures can be found in Appendix A.3.

The hypothesis is that we ought to see higher differential entropy for distributional uncertainty for the digits stemming from a wide array of fonts present in NotMNIST and SVHN as none of the fonts are handwritten. We additionally compare with FashionMNIST, to discern if models are capable of flagging anomalous inputs when deployed in a user-facing system.

We notice an abrupt loss in AUC scores on SVHN and FashionMNIST for 3 layers DCGP when transitioning from 50 to 100 inducing points. This reinforces the observed pathology that in most

Model	Hidden Layers	MNIST vs. NotMNIST AUC		MNIST vs. SVHN AUC		MNIST vs. FashionMNIST AUC	
Inducing Points		50	100	50	100	50	100
DCGP	1	0.71	0.69	**1.0**	**1.0**	**0.98**	0.98
DistDCGP	1	**0.87**	**0.95**	**1.0**	**1.0**	**0.98**	**1.0**
DCGP	2	0.64	0.59	0.97	0.99	0.94	0.92
DistDCGP	2	**0.97**	**0.84**	**1.0**	**1.0**	**1.0**	**0.98**
DCGP	3	0.64	0.70	0.97	0.89	0.94	0.84
DistDCGP	3	**0.95**	**0.86**	**1.0**	**1.0**	**0.96**	**0.92**

Table 3.2: OOD detection results. Performance of OOD detection based on distributional uncertainty differential entropy. Models with different inducing point numbers in hidden layers are trained on MNIST (normative data).

cases, increasing the number of inducing points results in loss of OOD capabilities. One notable exception is for discerning NotMNIST from MNIST, whereby increasing the number of inducing points actually helps (only in the case of 3 hidden layers). On NotMNIST we notice a substantial discrepancy between the two in favor of DistDCGP (see Table 3.2). Overall, DistDCGP manages to surpass or be on par with DCGP in any given comparison regardless of network architecture. Further figures showcasing differences in their distributional differential entropy histograms are provided in Appendix A.5.2.

3.6 Discussion

We have shown under a set of assumptions that total uncertainty can be correctly propagated under the DSVI framework introduced in Salimbeni and Deisenroth (2017) only in the case that the first order derivative of the predictive mean function attains maximum values at the point where previously identified OOD points get mapped to. Intuitively, total uncertainty for OOD points will be higher compared to in-distribution points, if the parametric component of the current layer's SVGP provides a diverse set of function samples when evaluated at points where the OOD point got mapped in the previous layer.

The collapse of distributional uncertainty as we increase the number of inducing points in the hidden layers is caused by the progressive mapping of data points which in lower layers were flagged as OOD to regions close to inducing points in the next layer in the case that in-distribution data points are mapped in intervals centred around zero in hidden layer space. This leads to the deactivation of $h(\cdot)$ (see equation 2.53, *i.e.*, non-parametric component in SVGP) in all layers, except the first. Hence, in these situations DGP rely just on the variance stemming from $g(\cdot)$ (see equation 2.54, *i.e.*, parametric component of SVGP). These aforementioned remarks are valid only for

zero-mean function DGP.

The PCA based projections stemming from the training data as introduced in Salimbeni and Deisenroth (2017) help to solve these aforementioned issues. However, these also come with a hidden cost, as the PCA projections will wildly extrapolate the regions of low distributional uncertainty in directions perpendicular to the eigenvectors in the case of high-dimensional datasets that are projected down in hidden layer space (see experiments on “banana" dataset as shown in Figure 3.8). Our proposed framework solves this problem as the Wasserstein-2 part of our hybrid kernel explicitly takes into account differences between the variance of hidden layer SVGPs. Intuitively, even if the predictive mean component of the SVGP is dominated by the PCA projection, which could coincide with the projection values of the training set, if the variance component is higher then the Wasserstein-2 distance will highlight this domain shift with higher distributional uncertainty. Besides this intuition, we have also provided theoretical guarantees that total uncertainty is higher for OOD points compared to in-distribution points for DDGP in the case that the inducing locations' variance is close in distribution to total uncertainty of in-distribution points at that hidden layer. This is a scenario which happens in practice as the variance of inducing locations will be optimized to match that of the in-distribution points. Future work should explore methods to explicitly enforce this match in distribution between inducing locations and in-distribution predictive distributions. Recent work (Ober and Aitchison 2021; Ustyuzhaninov et al. 2019) has proposed to propagate the initial inducing locations in input space through the hierarchy, thus obtaining predictive distributions for these “global" inducing points at each layer. Under this construction, the distribution match between inducing locations and in-distribution points will be guaranteed at each layer in DDGP. This remains as a prospective research avenue.

Besides theoretical guarantees on correct propagation of distributional uncertainty in DDGP, we have also shown that DDGP have theoretical guarantees to better guard against *feature-collapse*. Moreover, the Wasserstein-2 distance based on predictive moments will cause DDGP to be smoother compared to DGP (see Figure 3.9). Due to all these attractive properties of DDGP, it is perhaps unsurprising that on OOD detection tasks our proposed model has managed to surpass baseline models.

Besides obtaining better OOD detection results, another important aspect is to obtain well calibrated predictive distributions at testing time. From our UCI datasets experiments we have seen that DDGP outperform their DGP counterparts in terms of predictive log-likelihood on testing points regardless of the number of layers. We hypothesize that this is primarily caused by the fact that DDGP have two pathways in its mapping between hidden layers: a stochastic one represented by the process of sampling and the Euclidean kernel and a deterministic one, which comprises the underlying Gaussian distributions and the Wasserstein-2 distance based kernel (see equations (3.6) and (3.7)). The former has the role of maintaining the stochastic nature of hidden layers (implicitly maintaining the resulting model as a DGP, as opposed to a deterministic kernel-based deep

embedding akin to the encoder network present in DKL) and to introduce correlations between data points, whereas the latter pathway was motivated due to its ability to discern between data points which were previously identified as outliers or inliers. However, the deterministic pathway might have another role, that is to transfer information throughout the hierarchy without the risk of losing valuable information through sampling, thus potentially resulting in faster convergence properties.

Hidden layer features in very wide (*i.e.*, high-dimensional hidden layers) DGP using isotropic kernels can collapse to a thin hyper-sphere. In Appendix A.4 we offer more details on this pathology, also showing that hybrid kernels can guard against it.

In this chapter we have used Wasserstein-2 based kernels operating on distributions due to the simplicity of their analytical formula for Gaussian distributions, one that directly uses the variance terms of the two distributions to be compared, hence enhancing OOD detection capabilities (see Figure 3.4 for illustration). Besides this, kernels using OT are the default method for GP operating on distributions (Bachoc et al. 2023). However, other methods could have been applied such as Probability Product Kernels (Jebara et al. 2004) or maximum mean discrepancy (Gretton et al. 2012; Modeste and Dombry 2022). These pursuits remain viable research topics for future work.

In conclusion, we have shown that DDGP are better suited for outlier detection on both toy and real datasets, while also showing slight improvements at testing time in medium and large scale datasets.

Chapter 4

Distributional Gaussian Processes Layers

This chapter is based on the following publication:

(a) **Popescu, S. G.**, Sharp, D. J., Cole, J. H., Kamnitsas, K., and Glocker, B. (2021). "Distributional Gaussian Process Layers for Outlier Detection in Image Segmentation". In: *International Conference on Information Processing in Medical Imaging*. Springer, pp. 415–427 – (Popescu et al. 2021)

The code for all experiments is available at https://github.com/SebastianPopescu/DistGP_Layers.

We propose a parameter efficient Bayesian layer for hierarchical convolutional Gaussian Processes that incorporates Gaussian Processes operating in Wasserstein-2 space to reliably propagate uncertainty. This directly replaces convolving Gaussian Processes with a distance-preserving affine operator on distributions. Our experiments on brain tissue-segmentation show that the resulting architecture approaches the performance of well-established deterministic segmentation algorithms (U-Net), which has not been achieved with previous hierarchical Gaussian Processes. Moreover, by applying the same segmentation model to out-of-distribution data (*i.e.*, images with pathology such as brain tumors), we show that our uncertainty estimates result in out-of-distribution detection that outperforms the capabilities of state-of-the-art OOD detection models and reconstruction-based approaches that learn normative distributions.

4.1 Introduction

Deep learning methods have achieved state-of-the-art results on a plethora of medical image tasks, such as segmentation and clinical risk assessment (Imai et al. 2020; Tang 2019; Zhou et al. 2021). However, their application in clinical settings remains challenging due to issues pertaining to lack of reliability and miscalibration of confidence estimates. Reliably estimating uncertainty in predictions is also of vital interest in adjacent machine learning fields such as reinforcement learning, to guide exploration, or in active learning, to guide the selection of data points for the next iteration of labelling.

As we previously discussed in section 1.1, distributional uncertainty represents a recent addition to the uncertainty quantification toolkit. Unsurprisingly, most research pertaining to incorporating uncertainty into medical image segmentation has gravitated around modelling inter-rater variability and the inherent aleatoric uncertainty associated to the dataset, which can be caused by noise in the data acquisition process or inter-class ambiguities, alongside modelling the uncertainty in parameters (Czolbe et al. 2021). However, less focus has been placed on how models behave when processing unexpected inputs which differ from the characteristics of the training data (*i.e.*, quantifying distributional uncertainty). Such outliers or out-of-distribution samples, could possibly lead to deleterious effects in healthcare applications, where predictive models may encounter data that is corrupted or from patients with diseases that the model is not trained for (Curth et al. 2019; Mårtensson et al. 2020).

OOD detection in medical imaging has been mostly approached through the lens of reconstruction-based techniques involving some form of encoder-decoder network trained on normative datasets (Chen et al. 2019, 2021). Conversely, we focus on enhancing task-specific models (*e.g.*, a segmentation model) with reliable uncertainty quantification that enables outlier detection, based on the *distance-aware* property introduced in section 1.2. This will represent the primary goal of this chapter.

An automated method that achieves higher accuracy, but with the downside of inaccurately capturing uncertainty can prove to be dangerous (Jiang et al. 2012). With that in mind, standard DNN, despite their high predictive performance, often show poor calibration between confidence estimates and prediction accuracy, as estimated by Expected Calibration Error (Guo et al. 2017), when processing unseen samples that are not from the data manifold of the training set (*e.g.*, in the presence of pathology). To alleviate this, Bayesian approaches that assign posteriors over weights or in function space have been proposed (Wilson and Izmailov 2020). However, either assigning priors on weights or in function space does not necessarily lend itself to reliable OOD detection capabilities by virtue of inspecting the predictive variance of the model as was shown in Henning et al. (2021). The authors argue that infinite-width networks, trained via the Neural Network Gaussian Process (NNGP) kernel (Lee et al. 2017), are not reliable for OOD detection since they

show that the associated NNGP kernel is not correlated with distances between objects in input space. This loss of *distance-awareness* after encoding data has catastrophic effects on OOD detection, as we will soon see. Similarly, Foong et al. (2019) describe a limitation in the expressiveness of the predictive uncertainty estimate given by mean-field variational inference (MFVI) when applied as the inference technique for Bayesian Neural Networks (BNNs). Concretely, MFVI fails in offering reliable uncertainty estimates in regions between well-separated clusters of data, which the authors coin as *in-between* uncertainty, with potentially catastrophic consequences for active learning, Bayesian optimisation and OOD detection. This over-confidence in regions between data clusters can be caused by the fact that the mean field assumption can in some cases severely restrict the complexity of uncertainty estimates a BNN can express in function space. The authors prove that strong dependencies are required between parameters so as to both fit the data regions and exhibit increased predictive uncertainty in the *in-between* regions (see appendices C and D in Foong et al. (2019)).

In this chapter we follow an alternative approach, using GP as the building blocks for deep Bayesian networks as we have previously discussed in section 1.3. The use of GP for image classification has garnered interest in the past years. Hybrid approaches, whereby a DNN's embedding mechanism is trained end-to-end with a GP as the classification layer, were the first attempts to unify the two approaches (Bradshaw et al. 2017). The first convolutional kernel was proposed in Wilk et al. (2017), constructed by aggregating patch response functions, which they authors have coined as Convolutional GP (CGP). This type of GP was placed on top of feed forward layers governed by GP applied in a convolutional manner, with promising improvements in accuracy compared to their shallow counterpart (Blomqvist et al. 2018). We expand on the latter work, by introducing a simpler convolutional mechanism, which does not require convolving GP at each layer and hence alleviates the computational cost of optimizing over inducing points' locations residing in high dimensional spaces alongside the issues brought upon by multi-output GP. We propose a plug-in Bayesian layer more amenable to CNN architectures. More concretely, we seek to replace each individual component of a standard convolutional layer in convolutional neural networks (CNN), respectively the convolved filters and the non-linear activation function. Firstly, we impose constraints on the filter such that we have an upper bound on distances after the convolution with regards to distances between the same objects beforehand. This will ensure that objects which were close in previous layers will remain close going forward, which as we shall see later on is a fundamental property for reliable OOD detection. Moreover, directly using convolved filters as opposed to convolved GP (Blomqvist et al. 2018) solves the issue with optimizing high-dimensional inducing points' locations alongside introducing a simpler mechanism by which we can introduce correlations between channels (Nguyen, Bonilla, et al. 2014). Secondly, we replace the element-wise non-linear activation functions with Distributional Gaussian Processes (DistGP) (Bachoc et al. 2017) used in a one-to-one mapping manner, essentially acting as a non-parametric activation function. In this chapter we will show that our proposed module is also suited for OOD detection on both toy/image data

and biomedical scans.

Related work. Research into Bayesian models has focused on a separation of uncertainty into two different types, aleatoric and epistemic. The former is irreducible, given by noise in the data acquisition process and has been considered in medical image segmentation (Monteiro et al. 2020), whereas the latter can be reduced by providing more data during model training. This has also been studied in segmentation tasks (Nair et al. 2020). Previous work proposed to account for the uncertainty in the learned model parameters using an approximate Bayesian inference over the network weights (Kendall et al. 2015). However, it was shown that this method may produce samples that vary pixel by pixel and thus may not capture complex spatially correlated structures in the distribution of segmentation maps. The probabilistic U-Net (Kohl et al. 2018) produces samples with limited diversity due to the fact that stochasticity is introduced in the highest resolution level of the U-Net (Ronneberger et al. 2015). To solve this issue, Baumgartner et al. (2019) introduce a hierarchical structure between the different levels of the U-Net, hence introducing stochasticity at each level. Another improvement on the Probabilistic U-Net comes by adding variational dropout (Kingma et al. 2015) to the last layer to gain epistemic uncertainty quantification properties (Hu et al. 2019). All the models previously introduced relied on multiple annotations of the images with the intended goal of capturing uncertainty in annotations via sampling from some form of latent variables which encode information about the whole image at varying scales of the U-Net. However, none of these previous works test how their models behave in the presence of outliers, thereby do not consider techniques for quantifying distributional uncertainty, that increases away from the training data manifold (see section 1.1), which we use here as a measure of OOD detection. To find similar metrics for OOD detection we explored general OOD literature for models which can be used for image segmentation. Variations of classical one-versus-all models have been adapted to neural networks (Franchi et al. 2020; Padhy et al. 2020). The closest work that we could find to our proposed approach uses a deep network as a feature extractor for a RBF network (Amersfoort et al. 2020). The authors quantify epistemic uncertainty as the L2 distance between a given data point and centroids corresponding to different classes, much alike the RBF kernels and the inducing point approach seen in SGP. The authors enforce Lipschitz smoothing of their feature extractor through a double gradient penalty, to maintain the L2 norm of the Jacobian close to the Lipschitz constant. Through this technique, the authors make the case that their proposed bi-Lipschitz network does not suffer from *feature collapse*, which is the pathological scenario whereby almost all data from the training set manifold (in some cases including OOD manifolds) gets mapped to a small finite set of values, with detrimental effects for OOD detection capabilities (see section 1.2).

Contributions. To summarize, this chapter makes the following advancements:

- We introduce a Bayesian layer that combines convolved affine operators that are upper bounded in Wasserstein-2 space and DistGP as "activation functions", which results in an expressive

non-parametric convolutional layer with Lipschitz continuity and reliable uncertainty quantification.

- We derive theoretical requirements for the model to not suffer from *feature collapse*, with additional empirical results to support the theory.
- We demonstrate that a GP-based convolutional architecture can achieve competitive results in segmentation tasks in comparison to a U-Net.
- We show improved OOD detection results on both general OOD detection tasks and on medical images compared to previous OOD approaches such as reconstruction-based models.

4.2 Deep Wasserstein Kernel Learning

Shift towards single-pass uncertainty quantification. Early methods for uncertainty quantification in BDL have focused on estimating the variance of samples from different sub-models, such as in using dropout (Gal and Ghahramani 2016a), Deep Ensembles (Lakshminarayanan et al. 2017) or in sampling posterior network weights from a hypernetwork (Pawlowski et al. 2017). This results in slow uncertainty estimation at testing time, which can be critical in high-risk domains where speed is of essence (*e.g.*, self-driving cars). Recent work in OOD detection has focused on estimating proxies for distributional uncertainty in a single-pass, such as in bi-Lipschitz regularized feature extractors for GP (Amersfoort et al. 2021; Liu et al. 2020a) or in parametrizing second-order uncertainty via neural networks within the framework of evidential learning (Amini et al. 2019; Charpentier et al. 2020). With this shift towards single-pass uncertainty quantification, DDGP and the hybrid kernel introduced in subsection 3.4.2 are no longer appropriate since they involve sampling the features at each hidden layer. In the next subsection we detail a deterministic variant of DDGP that still preserves correlations between data points in the hidden layers.

4.2.1 Generative Process

We now write the generative process of this new probabilistic framework entitled Deep Wasserstein Kernel Learning (DWKL) for 2 layers:

$$p(F_1) = \mathcal{N}\left(\mathrm{PCA}(F_0), \mathrm{diag}\left[K_{ff}\right]\right) \tag{4.1}$$

$$p(F_2) = \mathcal{N}\left(0, k^{W_2}\left(p(F_1), p(F_1)\right)\right) \ , \tag{4.2}$$

where for simplicity we consider $F_0 = X$ and by $\mathrm{PCA}(F_0)$ we denote the PCA mean function for the corresponding points in input space. Due to the introduction of PCA mean functions, data

points in the hidden layer are now correlated. To make this clear, we can explicitly calculate it:

$$p\begin{pmatrix}F_{2,i}\\F_{2,j}\end{pmatrix} \sim \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}\sigma_2^2 \exp -\frac{-W_2^2(\mu_i,\mu_i)}{l^2} & \sigma_2^2 \exp -\frac{-W_2^2(\mu_i,\mu_j)}{l^2}\\ \sigma_2^2 \exp -\frac{-W_2^2(\mu_j,\mu_i)}{l^2} & \sigma_2^2 \exp -\frac{-W_2^2(\mu_j,\mu_j)}{l^2}\end{bmatrix}\right) \tag{4.3}$$

$$\sim \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}\sigma_2^2 & K_{i,j}^{W_2}\\ K_{j,i}^{W_2} & \sigma_2^2\end{bmatrix}\right), \tag{4.4}$$

where $\mu_i \sim \mathcal{N}\left(\mathrm{PCA}(F_{0,i}),\ \sigma_1^2\right)$ and $\mu_j \sim \mathcal{N}\left(\mathrm{PCA}(F_{0,j}),\ \sigma_1^2\right)$. If the PCA embeddings of x_i and x_j are different, then the Wasserstein-2 distance will be different than zero, hence introducing correlations.

4.2.2 Evidence lower bound

Deep Kernel Learning (DKL) (Wilson et al. 2016) is defined as a shallow GP with the input encoded by a neural network:

$$p(Y, F_L, U_L) = \underbrace{p(Y|F_L)}_{\text{likelihood}} \underbrace{p(F_L|U_L; Z_{L-1}, \mathrm{Enc}(X))p(U_L)}_{\text{prior}}, \tag{4.5}$$

where $\mathrm{Enc}(X)$ represents the input passed through a neural network encoder, which provides a deterministic transformation of the data that is then fed into a SVGP operating on Euclidean data (*i.e.*, using standard squared exponential kernel).

We diverge from this approach by using a succession of DistGP with Wasserstein-2 kernels as the encoder network, hence our transformed input is given by a Gaussian distribution $q(F_{L-1})$. Using the first two moments of the penultimate layer, we introduce a DistGP so as to obtain the final predictions. The conditional equation for DistGP at arbitrary layer $l \geq 2$ is written as:

$$p(F_l \mid U_l; Z_{l-1}, F_{l-1}) = \mathcal{N}(F_l \mid K_{fu}^{W_2} {K_{uu}^{W_2}}^{-1} U, K_{ff}^{W_2} - Q_{ff}^{W_2}), \tag{4.6}$$

where we have distributional inducing points $Z_{l-1} \sim \mathcal{N}(Z_{l-1} \mid M_{Z_{l-1}}, \Sigma_{Z_{l-1}})$ and uncertain input stemming from previous layer $F_{l-1} \sim \mathcal{N}\left(F_{l-1} \mid \tilde{U}_{l-1}(F_{l-2}), \tilde{\Sigma}_{l-1}(F_{l-2})\right)$. For computational reasons we take both $\tilde{\Sigma}_{l-1}(F_{l-2})$ and Σ_{Z_l} to be diagonal matrices. The first layer is governed by a standard SVGP operating on Euclidean data, hence for $l = 1$ we have $q(F_1) \sim \mathcal{N}\left(F_1 \mid \tilde{U}_1(x), \tilde{\Sigma}_1(x)\right)$, which is the standard predictive equation for SVGP as given in equations (2.41) and (2.42).

The joint density prior of DWKL is given as:

$$\underbrace{p(Y \mid F)}_{\text{likelihood}} \underbrace{p(F_L \mid U_L; Z_{L-1}, Enc(X)) \prod_{l=1}^{L} p(U_l)}_{\text{prior}} , \tag{4.7}$$

where in our case $Enc(X) \sim \mathcal{N}\left(\tilde{U}_{L-1}(F_{L-2}), \tilde{\Sigma}_{L-1}(F_{L-2})\right)$ that acts as the uncertain input for the final layer DistGP that maps to output space. We introduce a factorized posterior between layers and dimensions $q(F_L, \{U_l\}_{l=1}^L) = p(F_L|U_L; Z_{L-1}) \prod_{l=1}^L q(U_l)$, where $q(U_l)$ is taken to be a multivariate Gaussian with mean m_{U_l} and variance S_{U_l}. This gives the DWKL variational lower bound:

$$\mathcal{L}_{DKWL} = \mathbb{E}_{q(F_L, \{U_l\}_{l=1}^L)} p(Y|F_L) - \sum_{l=1}^{L} \mathrm{KL}\left[q(U_l) \| p(U_l)\right] , \tag{4.8}$$

where $q(F_L) = \mathcal{N}(\tilde{U}_L(Enc(X)), \tilde{\Sigma}_L(Enc(X)))$. For $1 \leq l \leq L-1$, F_l act as features for the next kernel as opposed to random variables that need to be integrated out (as in the case of DGP or DDGP). We provide the pseudocode of the previously mentioned operations (see Algorithm 1).

Algorithm 1: Deep Wasserstein Kernel Learning

Input: Euclidean data $X = F_0$
First layer is standard sparse variational GP
Variational Parameters: $U_1 \sim \mathcal{N}(m_{U_1},\ S_{U_1})$
Inducing Points: Euclidean space Z_0
$q(F_1) = \mathcal{N}(F_1 \mid K_{fu}^{SE} {K_{uu}^{SE}}^{-1} m_{U_1}, K_{ff}^{SE} - K_{fu}^{SE} {K_{uu}^{SE}}^{-1} (K_{uu}^{SE} - S_{U_1}) {K_{uu}^{SE}}^{-1} K_{uf}^{SE}$
for $l = 2$ **to** L **do**
  Hidden layers are distributional sparse variational GP
  Variational Parameters: $U_l \sim \mathcal{N}(m_{U_l},\ S_{U_l})$
  Distributional Inducing Points: $Z_{l-1} \sim \mathcal{N}(M_{Z_{l-1}},\ \Sigma_{Z_{l-1}})$
  Compute $K_{fu}^{W_2}$**:** $\sigma_l^2 \exp \sum_{d=1}^{D_l} \frac{-W_2^2(q(F_{l-1}[:,d]), Z_{l-1}[:,d])}{l_{l,d}^2}$
  Compute $K_{uu}^{W_2}$**:** $\sigma_l^2 \exp \sum_{d=1}^{D_l} \frac{-W_2^2(Z_{l-1}[:,d], Z_{l-1}[:,d])}{l_{l,d}^2}$
  $q(F_l) = \mathcal{N}(F_l \mid K_{fu}^{W_2} {K_{uu}^{W_2}}^{-1} m_{U_l}, K_{ff}^{W_2} - K_{fu}^{W_2} {K_{uu}^{W_2}}^{-1} \left[K_{uu}^{W_2} - S_{U_l}\right] {K_{uu}^{W_2}}^{-1} K_{uf}^{W_2}$
end for
Maximize ELBO: $\mathbb{E}_{q(F_L), \{q(U_l\}_{l=1}^L)} p(Y|F_L) - \sum_{l=1}^L \mathrm{KL}\left[q(U_l) \| p(U_l)\right]$

4.3 Distributional GP Layers

Integrating GP in convolutional architectures. The standard choice of using GP in a convolutional manner is the Deep Convolutional Gaussian Processes model (Blomqvist et al. 2018), whereby a convolutional GP (Wilk et al. 2017) is used on top of a succession of feed-forward SVGP

layers which process data in a convolutional manner similar to standard convolutional layers. This formulation does come with some drawbacks as we highlighted in subsection 1.3, namely the issue of devising high-dimensional multi-output GP (Wilk et al. 2020a) and optimizing high-dimensional inducing point locations in the case of using convolutional architectures with high-dimensional channels. In this section we introduce an alternative to this framework for integrating GP in a convolutional architecture, one that is more amenable to modern convolutional architectures. We commence by describing this probabilistic module, subsequently computing conditions for Lipschitz continuity while also providing theoretical requirements for *feature-collapse* to occur, alongside investigating if over-correlation in the representation learning layer is implicitly favoured via our loss function. Lastly, we adapt standard CNN-specific layers (*e.g.*, average pooling) to be suitable for our proposed probabilistic framework.

4.3.1 Module architecture

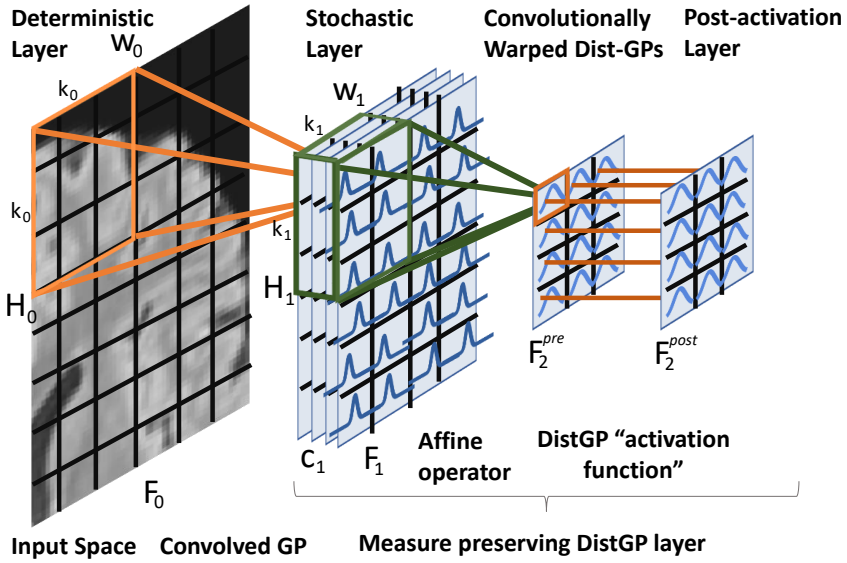


Figure 4.1: Schematic of measure-preserving DistGP layer. Sparse variational GP is convolved on input data to obtain first hidden layer. Lipschitz constrained affine operator is convolved on stochastic layer, propagating both mean and variance to obtain the pre-activation of the second hidden layer. Distributional GP is applied element-wise to introduce non-linearities and to propagate distributional uncertainty in the post-activation of the second hidden layer.

For ease of notation and graphical representation we describe the case of the input being a 2D image, with no loss of generality. We denote the image's representation $F_l \in \mathbb{R}^{H_l, W_l, C_l}$ with width W_l, height H_l and C_l channels at the l-th layer of a multi-layer model. F_0 is the image. Consider a square kernel of size $k_l \times k_l$. We denote with $F_l^{[p,k_l]} \in \mathbb{R}^{k_l, k_l, C_l}$ the p-th patch of F_l, which is the area of F_l that the kernel covers when overlaid at position p during convolution (e.g. orange square for a 3×3 kernel in Figure 4.1). We introduce the convolved $GP_0 : F_0^{[p,k_0]} \to \mathcal{N}(m, k)$ with $Z_0 \in \mathbb{R}^{k_0, k_0, C_0}$ to be the SGP operating on the Euclidean space of patches of the input image in a similar fashion to the layers introduced in Blomqvist et al. (2018). For $1 \leq l \leq L$ we introduce

affine operators $A_l \in \mathbb{R}^{k_l,k_l,C_{l-1},C_{l,pre}}$ which are convolved on the previous stochastic layer in the following manner:

$$m(F_l^{pre}) = \text{Conv}_{2\text{D}}(m(F_{l-1}), A_l) \tag{4.9}$$

$$v(F_l^{pre}) = \text{Conv}_{2\text{D}}(v(F_{l-1}), A_l \odot A_l)\,, \tag{4.10}$$

where $\odot$ represents the Hadamard product. The affine operator is sequentially applied on the mean, respectively variance components of the previous layer F_{l-1} so as to propagate the Gaussian distribution to the next pre-activation layer F_l^{pre}. To obtain the post-activation layer, we apply a $DistGP_l : F_l^{pre,[p,1]} \rightarrow \mathcal{N}(m,k)$ in a many-to-one manner on the pre-activation patches to arrive at F_l^{post}. Figure 4.1 depicts this new module, entitled "Measure preserving DistGP" layer with pseudo-code offered in Algorithm 2. In Blomqvist et al. (2018) the convolved GP is used across the entire hierarchy, thereby inducing points are in high-dimensional space ($k_l^2 * C_l$). In our case, the convolutional process is replaced by an inducing points free affine operator, with inducing points in low-dimensional space ($C_{l,pre}$) for the DistGP activation functions. The affine operator outputs $C_{l,pre}$, which is taken to be higher than the associated output space of DistGP activation functions C_l. Hence, the affine operator can cheaply expand the channels, in contrast to the layers in Blomqvist et al. (2018) which would require high-dimensional multi-output GP. We motivate the preservation of distance in Wasserstein-2 space in the following section. Previous research has highlighted the importance of having an upper bound on $\|h(x_1) - h(x_2)\|_h \leq L_{upper}\|x_1 - x_2\|_x$, as it ensures a certain degree of robustness towards adversarial examples, since it prevents the hidden forward mappings from being overly sensitive to the conceptually meaningless perturbations in input space (Jacobsen et al. 2018; Sokolić et al. 2017; Weng et al. 2018). Conversely, the lower bound $\|h(x_1) - h(x_2)\|_h \geq L_{lower}\|x_1 - x_2\|_x$ ensures that the forward mappings do not become invariant to semantically meaningful changes in the input Amersfoort et al. (2020).

4.3.2 Imposing Lipschitz Conditions in Convolutionally Warped DistGP

If a sample is identified as an outlier at certain layer, respectively being flagged with high variance, in an ideal scenario we would like to preserve that status throughout the remainder of the network. As the kernels operate in Wasserstein-2 space, the distance of a data point's first two moments with respect to inducing points is vital. Hence, we would like our network to vary smoothly between layers, so that similar objects in previous layers get mapped into similar spaces in the Wasserstein-2 domain. In this section, we accomplish this by quantifying the *"Lipschitzness"* of our "Measure preserving DistGP" layer and by imposing constraints on the affine operators so that they preserve distances in Wasserstein-2 space.

Proposition 4.3.1. *For a given DistGP F and a Gaussian distribution $\mu \sim \mathcal{N}(m_1, \Sigma_1)$ to be the centre of an annulus $B(x) = \{\nu \sim \mathcal{N}(m_2, \Sigma_2) \mid 0.125 \leq \frac{W_2(\mu,\nu)}{l^2} \leq 1.0$ and choosing any ν inside the ball we have the following Lipschitz bounds: $W_2(F(\mu), F(\nu)) \leq LW_2(\mu, \nu)$, where*

Algorithm 2: Distributional Gaussian Processes Layers

Input: Euclidean data $X = F_0 \in \mathbb{R}^{H_0, W_0, C_0}$
First layer is convolved sparse variational $GP_0 : F_0^{[p,k_0]} \to F_1^{[p]}$
Variational Parameters: $U_1 \sim \mathcal{N}(m_{U_1},\ S_{U_1})$
Inducing Points: Euclidean space $Z_0 \in \mathbb{R}^{k_0, k_0, C_0}$
$q(F_1) = \mathcal{N}(F_1 \mid K_{fu}^{SE} {K_{uu}^{SE}}^{-1} m_{U_1}, K_{ff}^{SE} - K_{fu}^{SE} {K_{uu}^{SE}}^{-1} (K_{uu}^{SE} - S_{U_1}) {K_{uu}^{SE}}^{-1} K_{uf}^{SE}$
for $l = 2$ **to** L **do**
  Affine operators: $A_l \in \mathbb{R}^{k_l, k_l, C_{l-1}, C_{l,pre}}$
  $m(F_l^{pre}) = \text{Conv}_{2\text{D}}(m(F_{l-1}), A_l)$
  $v(F_l^{pre}) = \text{Conv}_{2\text{D}}(v(F_{l-1}), A_l \odot A_l)$
  Hidden layer activation functions are distributional sparse variational GP
  $DistGP_l : F_l^{pre,[p,1]} \to F_l^{post,[p,1]}$
  Variational Parameters: $U_l \sim \mathcal{N}(m_{U_l},\ S_{U_l})$
  Inducing Points: $Z_{l-1} \sim \mathcal{N}(M_{Z_{l-1}},\ \Sigma_{Z_{l-1}})$
  Compute $K_{fu}^{W_2}$**:** $\sigma_l^2 \exp \sum_{d=1}^{D_l} \frac{-W_2^2(q(F_{l-1}[:,d]), Z_{l-1}[:,d])}{l_{l,d}^2}$
  Compute $K_{uu}^{W_2}$**:** $\sigma_l^2 \exp \sum_{d=1}^{D_l} \frac{-W_2^2(Z_{l-1}[:,d], Z_{l-1}[:,d])}{l_{l,d}^2}$
  $q(F_l^{post}) = \mathcal{N}(F_l \mid K_{fu}^{W_2} {K_{uu}^{W_2}}^{-1} m_{U_l}, K_{ff}^{W_2} - K_{fu}^{W_2} {K_{uu}^{W_2}}^{-1} \left[K_{uu}^{W_2} - S_{U_l}\right] {K_{uu}^{W_2}}^{-1} K_{uf}^{W_2}$
end for
Maximize ELBO: $\mathbb{E}_{q(F_L, \{q(U_l\}_{l=1}^L)} p(Y \mid F_L) - \sum_{l=1}^{L} \text{KL}\left[q(U_l) \| p(U_l)\right]$

$L = (\frac{4\sigma^2}{l})^2 \left[\|K_{uu}^{-1} m\|_2^2 + \|K_{uu}^{-1}\left(K_{uu} - S\right) K_{uu}^{-1}\|_2\right]$ *and* l, σ^2 *are the lengthscales and variance of the kernel.*

A proof of this proposition can be found in Appendix B.1.1

Remark 4.3.1. *This theoretical result shows that DistGP "*activation functions*" have Lipschitz constants with respect to the Wasserstein-2 metric in both output and input domain. This will ensure that the distance between previously identified outliers and inliers will stay constant if the Lipschitz constant is set to 1. However, it is worthy to highlight that we can only obtain locally Lipschitz continuous functions, given that we can only obtain Lipschitz constants for any Gaussian distribution* ν *inside the annulus* $B(x) = \{\nu \sim \mathcal{N}\left(m_2, \Sigma_2\right) \mid 0.125 \leq \frac{W_2(\mu,\nu)}{l^2} \leq 1.$ *with respect to the centre of the annulus,* μ*.*

We are now interested in finding Lipschitz constants for the affine operator A that gets convolved to arrive at the pre-activation stochastic layer.

Proposition 4.3.2. *We consider the affine operator* $A \in \mathbb{R}^{C,1}$ *operating in the space of multivariate Gaussian distributions of size C. Consider two distributions* $\mu \sim \mathcal{N}(m_1, \sigma_1^2)$ *and* $\nu \sim \mathcal{N}(m_2, \sigma_2^2)$*, which can be thought of as elements of a hidden layer patch, then for the affine operator function* $f(\mu) = \mathbb{N}(m_1 A, \sigma^2 A^2)$ *we have the following Lipschitz bound:* $W_2\left(f(\mu), f(\nu)\right) \leq L W_2\left(\mu, \nu\right)$*, where* $L = \sqrt{C}\|W\|_2^2$*.*

A proof of this proposition can be found in Appendix B.1.2.

Remark 4.3.2. *We denote the l-th layer weight matrix, computing the c-th channel by column matrix $A_{l,c}$. We can impose the Lipschitz condition to Eq. 4.9, 4.10 by having constrained weight matrices with elements of the form $A_{l,c} = \frac{A_{l,1}}{C^{\frac{1}{2}}\sqrt{\sum_{c=1}^{C} W_{l,c}^2}}$.*

4.3.3 *Feature-collapse* in DistGP Layers

In this subsection we delve deeper into the properties of DistGP layers from a function-space view. In light of recent interest into *feature collapse* (Amersfoort et al. 2020), which is the pathological phenomenon of having the representation layer collapse to a small finite set of values, with catastrophic consequences for OOD detection, we investigate what are the necessary conditions for our proposed network to collapse in feature space. Subsequently, we investigate if feature collapse is inherently encouraged by our loss function.

We commence by introducing notation conventions. We consider $\{u_l \in \mathbb{R}^{D_l}\}_{l=0:L}$ where D_l is the number of dimensions in the l-th layer of the hierarchy. We consider the following two functions $\Psi_l : u_{l-1} \to \mathcal{R}^{m_l}$ and $f_l : \mathcal{R}^{m_l} \to u_l$. To relate this notation to our construction of a DistGP layer introduced in section 4.3, m_l represents the number of dimensions of the warped GP (warping performed by affine deterministic layer; see dark green arrows in Figure 4.1). We denote by f_l to be the DistGP (mean function included) taking values in the space of continuous functions $C(u_l; \mathcal{R}^{m_l})$, which relates to the "activation function" construction from Figure 4.1. Then we have the following composition for a given DistGP layer:

$$u_l(x) = f_l\left(\Psi_l(u_{l-1})(x)\right) \ . \tag{4.11}$$

One can easily see that DWKL can be recovered by taking $\Psi_l = id$, instead of the affine embedding. The first layer prior $p\begin{pmatrix} u_1(x) \\ u_1(x^*) \end{pmatrix}$ is defined as follows:

$$\mathcal{N}\left(\begin{bmatrix} m_1(x) \\ m_1(x^*) \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & k^E\left(x, x^*\right) \\ k^E\left(x^*, x\right) & \sigma_1^2 \end{bmatrix}\right) \ . \tag{4.12}$$

We now define the prior post-activation layers $p\begin{pmatrix} u_l(x) \\ u_l(x^*) \end{pmatrix}$ for $l \geq 2$ in the following recursive manner:

$$\mathcal{N}\left(\begin{bmatrix} m_l(x) \\ m_l(x^*) \end{bmatrix}, \begin{bmatrix} \sigma_l^2 & k^{W_2}\left(\mu_{l-1}(x), \mu_{l-1}(x^*)\right) \\ k^{W_2}\left(\mu_{l-1}(x^*), \mu_{l-1}(x)\right) & \sigma_l^2 \end{bmatrix}\right) \ , \tag{4.13}$$

where $\mu_l\left(x\right) = \mathcal{N}\left(m_{l-1}(x)W_l,\ \sigma_{l-1}^2 W_l^2\right)$, where $m_{l+1}(\cdot) = \overline{\overline{m_1(\cdot)W_1}\cdots W_l}$ and $\overline{m_1(x)W_1}$

signifies having the Principal Component Analysis (PCA) mean function of the first layer multiplied by W_1 and averaged across its dimensions.

Proposition 4.3.3. *We assume μ_0 to be bounded on bounded sets almost-surely. If at each layer we have satisfied the following inequalities $D_l^2\langle\tilde{W}_l, \tilde{W}_l\rangle \leq 1$ and $\left[D_L * \langle\tilde{W}_L, \tilde{W}_L\rangle + \frac{\sigma_L^2}{2l_L^2}\right] \leq 1$, where D_l is the size of the l-th layer and $\tilde{W}_l$ represents a normalized version of the affine embedding W_l, we have the following result:*

$$P\left(\|u_n(x) - u_n(x^*)\|_2 \to 0\right) = 1 \tag{4.14}$$

A proof of this proposition can be found in Appendix B.1.3.

Remark 4.3.3. *If at each layer the following inequalities $D_l m_{l-1} D_{l-1} < \tilde{W}_l, \tilde{W}_l >\leq 1$ and $\left[m_l D_{L-1} * \langle\tilde{W}_L, \tilde{W}_L\rangle + \frac{\sigma_L^2}{2l_L^2}\right] \leq 1$ are satisfied then the network collapses to constant values. Intuitively, if the norm of W_l is not large enough, then it won't change the Gaussian random field too much. Furthermore, if σ_l^2 is larger, which translates in increased amplitude of the samples from the Gaussian random field, then the values will not collapse. As opposed to the hypothetical requirements for DGP (Dunlop et al. 2018), we can immediately notice that for DistGP layers there is no requirement for the kernel variance and lengthscales from intermediate layers, relying solely on the last layer hyperparameters. Lastly, we can notice that as the width of the stochastic layers is increased, alongside warped layers through affine embedding, the conditions are less likely to be satisfied.*

4.3.4 Over-correlation in latent space

Besides the issues pertaining to *feature collapse* explored in the previous subsection, we ought to be aware of another potential issue for our proposed backbone framework DWKL.

We remind ourselves that DKL consist of using a neural network to map inputs to an intermediate feature space, on which we use a GP. The network parameters can be considered as hyperparameters of the kernel, being optimized with respected to the marginal likelihood, as is the case with classic GP inference. From a Bayesian perspective, a higher marginal likelihood should translate to increased performance, being often claimed that optimizing the marginal likelihood will calibrate the complexity of the model, thus preventing overfitting Wilson and Nickisch (2015).

In Ober et al. (2021), the authors claim that this is indeed the case for models with a small amount of hyperparameters, also showing that for models with a large amount of hyperparameters such as the DKL, in some cases optimizing the marginal likelihood might encourage overfitting. The fundamental premise of the paper resides in the fact that due to the marginal likelihood, the neural network encoder might attempt to correlate all datapoints, as opposed to just the ones for which correlations are actually important. This phenomenon of "over-correlation" can be viewed through

the lense of *feature-collapse* as introduced in Amersfoort et al. (2020).

We commence by briefly introducing the main results and derivations from Ober et al. (2021), subsequently adapting them to the sparse GP scenario, which bears more resemblance to what would more likely be used in practice.

Over-correlation in Exact DKL: The claim that the marginal likelihood guards against over-fitting is based on the common decomposition of the log marginal likelihood into *data fit* and *complexity penalty* terms, more specifically for GP we have:

$$\log p(y) = \log \mathcal{N}\left(y \mid 0, K_{ff} + \sigma^2_{noise} I_n\right) \tag{4.15}$$

$$\propto \underbrace{-\frac{1}{2}\log \mid K_{ff} + \sigma^2_{noise} I_n \mid}_{\text{complexity penalty}} \underbrace{-\frac{1}{2} y^\top \left(K_{ff} + \sigma^2_{noise} I_n\right)^{-1} y}_{\text{data fit}}, \tag{4.16}$$

where we define the squared exponential kernel $k^{SE}(x_i, x_j) = \sigma_f^2 \exp\left[\sum_{d=1}^{D} -\frac{\left(x_{i,d} - x_{j,d}\right)^2}{2l_d^2}\right]$ for $\forall x_i, x_j \in \mathbb{R}^D$ and the hyperparameters to tune are given by the noise variance σ^2_{noise}, kernel variance σ_f^2 and lengthscales l_d^2.

The authors used the following reparametrization $\sigma^2_{noise} = \tilde{\sigma}^2_{noise}\sigma_f^2$, which results in $K_{ff} + \sigma^2_{noise} I_n = \sigma_f^2\left(\tilde{K}_{ff} + \tilde{\sigma}^2_{noise} I_n\right)$. The derivation proceeds by showing that if we are to differentiate the log marginal likelihood with respect to σ_f^2, we would obtain:

$$\frac{\mathrm{d}\ \log p(y)}{\mathrm{d}\ \sigma_f^2} = \frac{\mathrm{d}\ -\frac{N}{2}\log \sigma_f^2 - \frac{1}{2}\log\left|\tilde{K}_{ff} + \tilde{\sigma}^2_{noise} I_n\right| - \frac{1}{2\sigma_f^2} y^\top \left(\tilde{K}_{ff} + \tilde{\sigma}^2_{noise} I_n\right)^{-1} y}{\mathrm{d}\ \sigma_f^2} \tag{4.17}$$

$$= -\frac{N}{2\sigma_f^2} + \frac{1}{2\sigma_f^4} y^\top \left(\tilde{K}_{ff} + \tilde{\sigma}^2_{noise} I_n\right)^{-1} y\,. \tag{4.18}$$

Hence, if we are to set the derivative equal to zero, we would obtain the optimal value for $\sigma_f^2 = \frac{1}{N} y^\top \left(\tilde{K}_{ff} + \tilde{\sigma}^2_{noise} I_n\right)$. Plugging this value in the *data fit* term would result in it being equal to $\frac{N}{2}$, where N is the number of training points. Hence, once the model has converged, the only way the model can increase its log-likelihood score is by modifying the *complexity penalty* term, which can be broken up as follows:

$$\frac{1}{2}\log\left|K_{ff} + \sigma^2_{noise} I_n\right| = \frac{N}{2}\log \sigma_f^2 + \frac{1}{2}\log\left|\tilde{K}_{ff} + \tilde{\sigma}^2_{noise} I_n\right|\,. \tag{4.19}$$

The minimization of the *complexity penalty* term can be achieved in two ways: by minimization of σ_f or by minimization of $\left|\tilde{K}_{ff} + \tilde{\sigma}^2_{noise} I_n\right|$. However, there is not much benefit in minimizing σ_f as

this would have an adverse effect on the *data fit* term. The second way implies correlating the input points as much as possible while minimizing the impact this has on the *data fit* term. Hence, the only plausible phenomenon is to have high correlations values in K_{ff} so as to get a determinant close to 0, a phenomenon that the authors call *over-correlation*.

Over-correlation in SVI DKL: In the remainder of this subsection, we derive similar results to Ober et al. (2021) but in the sparse scenario (*i.e.*, using a SGPR instead of an exact GP within the DKL framework). We remind ourselves the collapsed bound (see equation (2.36)) introduced in Titsias (2009):

$$\mathcal{L}_{Titsias} = \log \mathcal{N}\left(y \mid Q_{ff} + \sigma^2_{noise} I_n\right) - \frac{1}{2\sigma^2_{noise}} \operatorname{Tr}\left[K_{ff} - Q_{ff}\right] \tag{4.20}$$

$$\propto \underbrace{-\frac{1}{2}\log \mid Q_{ff} + \sigma^2_{noise} I_n \mid}_{\text{complexity penalty}} \underbrace{-\frac{1}{2} y^\top \left(Q_{ff} + \sigma^2_{noise} I_n\right)^{-1} y}_{\text{data fit}} \underbrace{-\frac{1}{2\sigma^2_{noise}} \operatorname{Tr}\left[K_{ff} - Q_{ff}\right]}_{\text{trace term}} \tag{4.21}$$

$$\propto -\frac{N}{2}\sigma^2 - \frac{1}{2}\log \mid \tilde{Q}_{ff} + \tilde{\sigma}^2_{noise} I_n \mid - \frac{1}{2\sigma^2} y^\top \left[\tilde{Q}_{ff} + \tilde{\sigma}^2_{noise} I_n\right] y \tag{4.22}$$
$$- \frac{1}{2\tilde{\sigma}^2_{noise}} \operatorname{Tr}\left[\tilde{K}_{ff} - \tilde{Q}_{ff}\right] ,$$

where we have used again the following notation for kernel terms $k(\cdot,\cdot) = \sigma_f^2 \tilde{k}(\cdot,\cdot)$ and $\sigma^2_{noise} = \sigma_f^2 \tilde{\sigma}^2_{noise}$.

We adapt the previous derivation to our framework at hand, obtaining the following expression for $\frac{\mathrm{d}\mathcal{L}_{Titsias}}{\mathrm{d}\sigma_f^2}$:

$$= \frac{\mathrm{d} - \frac{N}{2}\sigma^2 - \frac{1}{2}\log\left|\tilde{Q}_{ff} + \tilde{\sigma}^2_{noise} I_n\right| - \frac{1}{2\sigma^2} y^\top \left[\tilde{Q}_{ff} + \tilde{\sigma}^2_{noise} I_n\right]^{-1} y - \frac{1}{2\tilde{\sigma}^2_{noise}}\left[\tilde{K}_{ff} - \tilde{Q}_{ff}\right]}{d\sigma^2} \tag{4.23}$$

$$= -\frac{N}{2\sigma^2} + \frac{1}{2\sigma^4} y^\top \left[\tilde{Q}_{ff} + \tilde{\sigma}^2_{noise} I_n\right]^{-1} y \,. \tag{4.24}$$

Hence, if we set the derivative to 0, then we obtain that $\sigma^2 = \frac{1}{N} y^\top \left[\tilde{Q}_{ff} + \tilde{\sigma}^2_{noise} I\right]^{-1} y$, which if we input it into the data fit term it results in $\frac{N}{2}$, similar to the non-sparse scenario analyzed in Ober et al. (2021). The difference between the sparse and non-sparse framework is that after convergence in the *data fit* term, the model now has to achieve *over-correlation* in both Q_{ff} and K_{ff}, so as to minimize $|Q_{ff} + \sigma^2_{noise} I_n|$ while also minimizing $\operatorname{Tr}\left[K_{ff} - Q_{ff}\right]$.

4.3.5 Pooling operations on stochastic layers

Previous work that dealt with combining GP with convolutional architectures Blomqvist et al. (2018), Dutordoir et al. (2020), and Kumar et al. (2018) have used in their experiments simple architectures involving a couple of stacked layers. In this paper, we propose to experiment with more modern architectures such as DenseNet Huang et al. (2017) or ResNet He et al. (2016). However, both these architectures include pooling layers such as average pooling, which for Euclidean data is a straightforward operation since we have a naturally induced metric. Since we are using stochastic layers that operate in the space of Gaussian distributions, this introduces some complications as it is not desirable to sample from the stochastic layers, subsequently applying the Euclidean space average pooling operation. Nevertheless, in the remainder of this subsection we show a simple method for replicating average pooling in Wasserstein space by using Wasserstein barycentres (Agueh and Carlier 2011).

We consider probability measures $\mu_1, ..., \mu_k$ and fixed weights $\theta_1, ..., \theta_k$ that are positive real numbers such that $\sum_{k=1}^{K} \theta_k = 1$. For $\nu \in \mathbb{P}_2(\mathbf{R}^d)$, where $\mathbb{P}_2$ is the set of Borel probabilities on $\mathbb{R}$ with finite second moment and absolutely continuous with respect to Lebesque measures, we consider the following functionals:

$$\mathbf{V}(\nu) = \sum_{k=1}^{K} \theta_k \mathbf{W}_2^2(\mu, \mu_k) \tag{4.25}$$

$$\mathbf{V}(\tilde{\mu}) = \min_{\mu \in \mathbb{P}_2} \mathbf{V}(\mu)\ , \tag{4.26}$$

where $\mathbf{V}(\tilde{\mu})$ is defined as the barycentre with respect to the Wasserstein-2 distance of the set of probabilities $\{\mu_1, ..., \mu_k\}$. Intuitively, barycentres can be seen as the equivalent of averaging in Euclidean space, while still maintaining the geometric properties of the distributions at hand.

Theorem 4.3.1 (Theorem 4.2. in Álvarez-Esteban et al. (2016))**.** *Assume $\Sigma_1, ..., \Sigma_K$ are symmetric positive semidefinite matrices, with at least one of them positive definite. We take $S_0 \in \mathbb{M}_{d\times d}^{+}$ and define:*

$$S_{n+1} = S_n^{-1/2} \left(\sum_{k=1}^{K} \theta_k (S_n^{1/2} \Sigma_k S_n^{1/2})^{1/2} \right)^2 S_n^{-1/2}\ . \tag{4.27}$$

If $\mathcal{N}(0,\ \Sigma_0)$ is the barycenter of $\mathcal{N}(0,\ \Sigma_1), ..., \mathcal{N}(0,\ \Sigma_K)$, then $W_2^2(\mathcal{N}(0,\ S_n), \mathcal{N}(0,\ \Sigma_0) \to 0$ as $n \to \infty$.

Remark 4.3.4. *In the case of computing the barycentre of univariate Gaussian measures, the iterative algorithm converges in one iteration to $\Sigma_0 = \left(\sum_{k=1}^{K} \theta_k \Sigma_k^{\frac{1}{2}} \right)^2$. This provides us with a deterministic and single step equation to downsample stochastic layers, where we can additionally calculate the mean of the barycentre by $\sum_{k=1}^{K} \theta_k m_k$, where $\{m_1, \cdots, m_K\}$ represent the first moments of the*

respective distributions.

4.4 DistGP Layer Networks & OOD detection

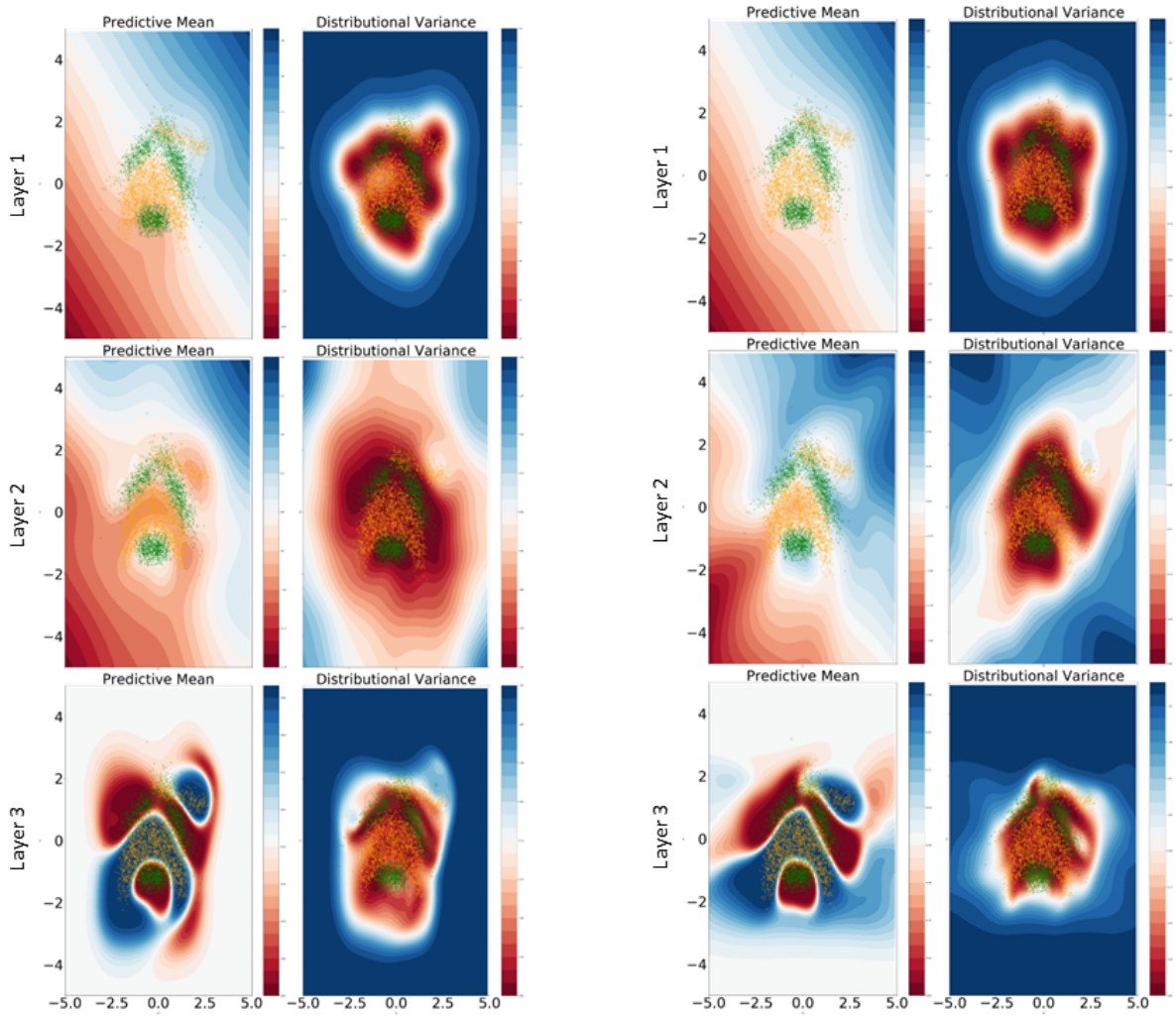
Influence of enforced Lipschitz condition. In this subsection we aim to visually assess if the Lipschitz condition imposed via Proposition 4.3.2 negatively influences the predictive capabilities. We use a standard neural network architecture with two hidden layers with 5 dimensions each, with the affine embeddings operations described in equations (4.9) and (4.10) being replaced by a non-convolutional dense layer. From Figure 4.2 we can notice that imposing a unitary Lipschitz constant does not result in the over-regularization of the predictive mean. A slight smoothing effect on the predictive mean can be noticed in output space. Moreover, for the Lipschitz constrained version we can discern a better fit of the data manifold in terms of distributional variance, with a noticeable difference in the second hidden layer.

Over-correlation in latent space. In this subsection we aim to understand whether the over-correlation phenomenon occurs for our model. We consider a standard neural network architecture with two hidden layers with 50 hidden units per layer. From Figure 4.3 we can notice that for DKL, the sparse framework does remove any unwanted over-correlations in the final hidden layer latent space. In the unconstrained model, there is a notion of locality in the final hidden layer latent space, albeit of a lower degree compared to the DKL model. With regards to OOD detection, of utmost importance is the fact that regions outside the training set manifold have a correlation value of 0. Perhaps unsurprisingly, introducing a unitary Lipschitz constraint had the effect of increasing the correlation in the latent space, alongside a smoother predictive mean.

Reliability of *in-between* uncertainty estimates. In this subsection, we are interested to test our newly introduced module in scenarios where *in-between* uncertainty can fail. For this we use the "snelson" dataset, with the training set taken to comprise the intervals between 0.0 and 2.0, respectively 4.0 and 6.5. Thereby, in an ideal scenario we would expect our model to offer high distributional uncertainty estimates between 2.0 and 4.0, which constitutes our *in-between* region. To benchmark our approach, we compare it to a collapsed SGPR as defined in Titsias (2009).

From Figure 4.4 we can observe that the behaviour is strikingly similar between a collapsed SGPR and a three layer DistGP-Layers network.

Reliability of epistemic uncertainty estimates. Epistemic uncertainty is responsible to detect regions of the input space where the variance in the model parameters, in this case of U, can be further reduced if we add more data points in said input regions. To test if our newly introduced module can provide reliable epistemic uncertainty estimates one can proceed to subsample



(a) DistGP-NN without Lipschitz constraint

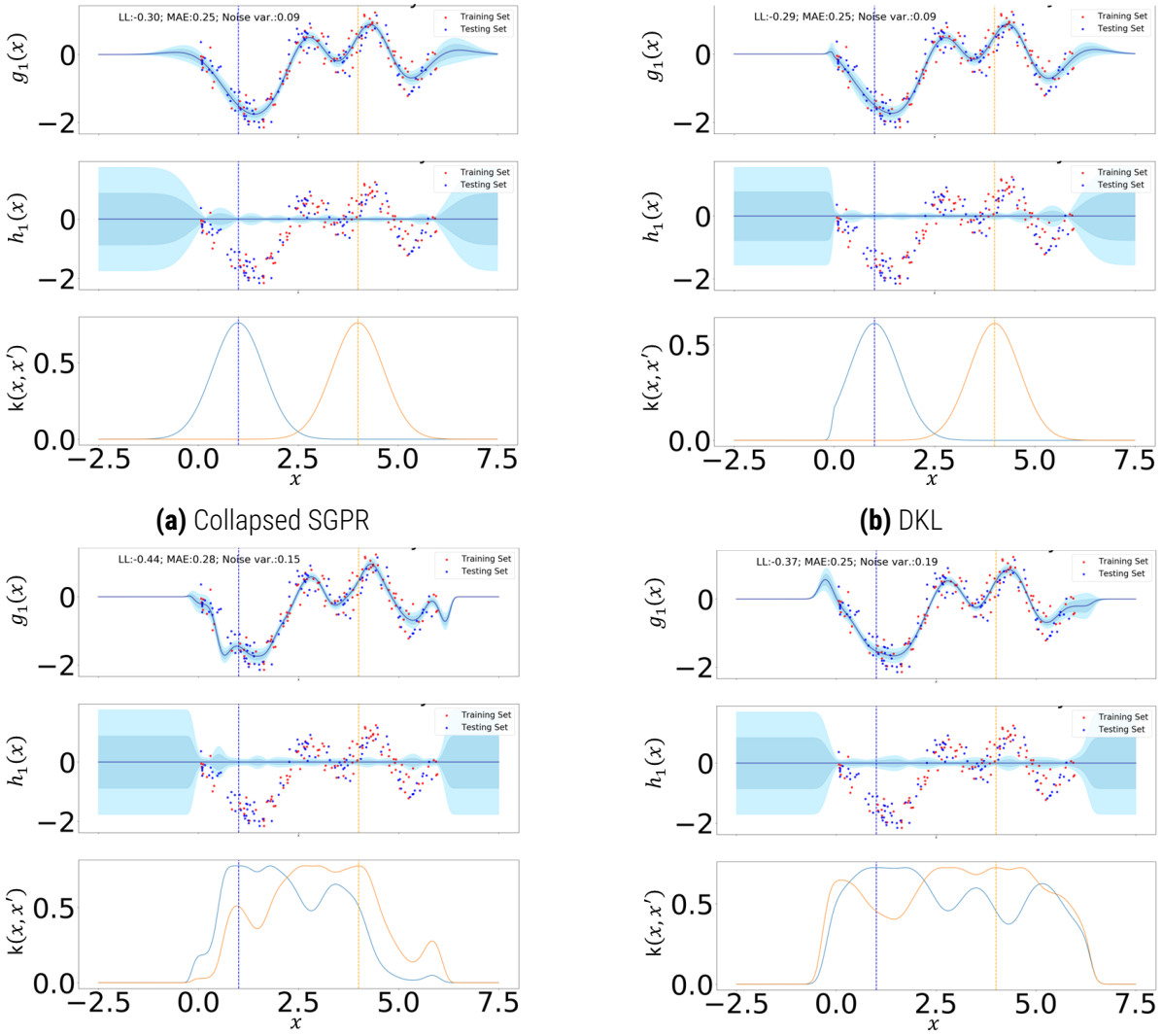
(b) DistGP-NN with Lipschitz constraint

Figure 4.2: Influence of Lipschitz continuity. Layer-wise predictive moments of DistGP-NN models (with or without unitary Lipschitz constraints) trained on toy binary classification dataset.

a dataset (as done in Figure 4.5 subsampling in the interval $[0, 2.5]$), with the intended effect being of an increase in epistemic uncertainty across the input region where we subsampled.

From Figure 4.5 we can see that despite the low number of training points, it did not result in overfitting, with our model exhibiting a relatively smooth predictive mean. Moreover, in comparison to Figure 4.3 we can notice that the epistemic uncertainty has substantially increased in the $[0, 2.5]$ interval.

MNIST and CIFAR10. We compare our approach on the standard image classification benchmarks of MNIST (Lecun et al. 1998) and CIFAR-10 (Krizhevsky 2009), which have standard training and test folds to facilitate direct performance comparisons. CIFAR-10 contains 50,000 training examples of RGB colour images of size 32×32 from 10 classes, with 5,000 images per class. We preprocess the images such that the input is normalized to be between 0 and 1. We compare our



(a) Collapsed SGPR

(b) DKL

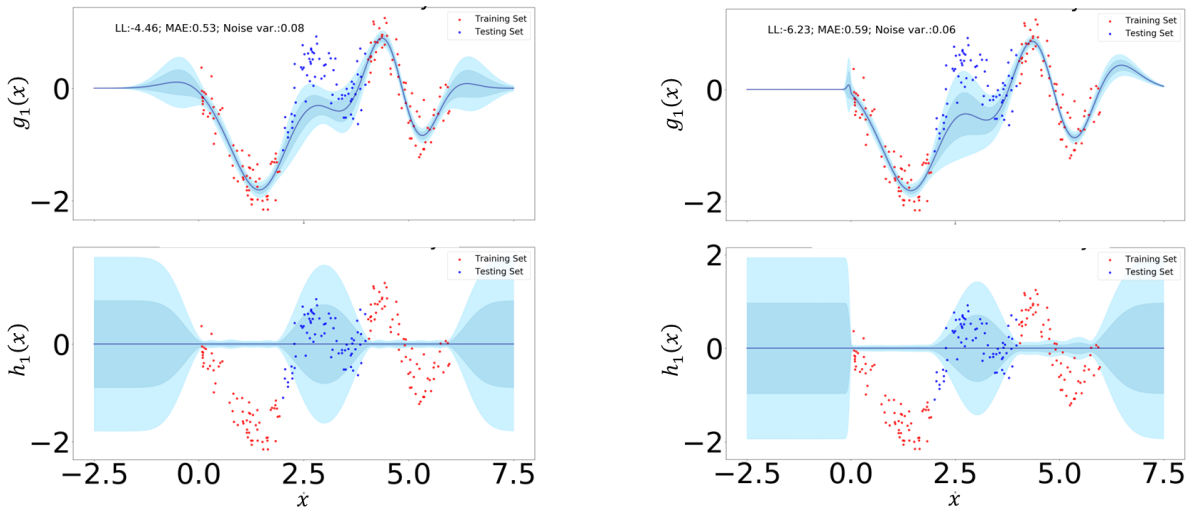
(c) DistGP-NN Not Constrained

(d) DistGP-NN Lipschitz Constrained

Figure 4.3: Correlation in representation learning layer. Top row: Predictive mean and variance of parametric part of SGP; Middle row: Predictive mean and variance of non-parametric part of SGP; **Bottom row:** Kernel evaluated across whole input span with respect to -2.0 (blue) and 2.0 (orange).

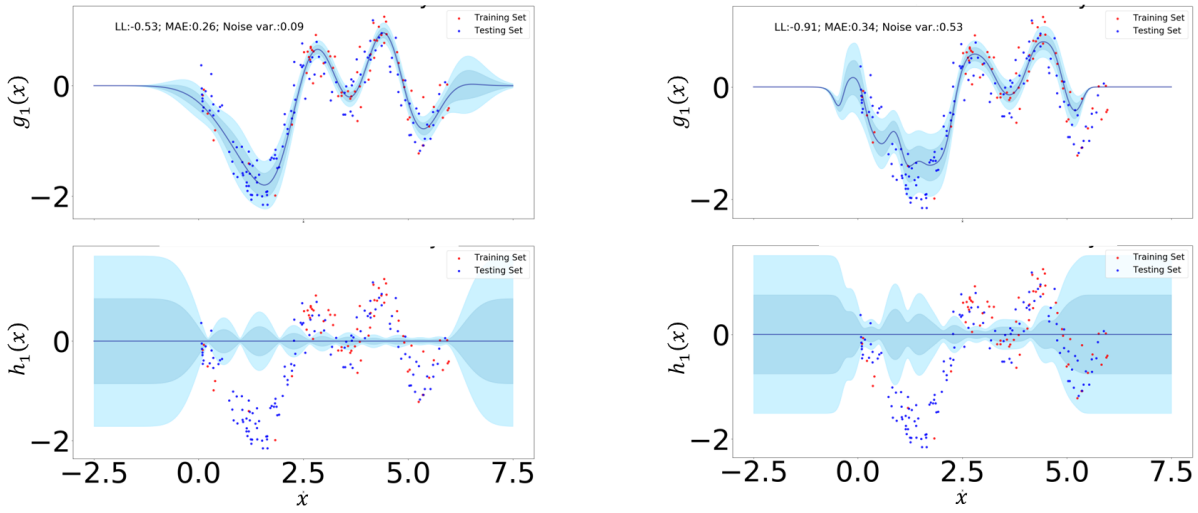
model primarily against the original shallow Convolutional Gaussian process (Wilk et al. 2017) and Deep Convolutional Gaussian Process (DeepConvGP) (Blomqvist et al. 2018). In terms of model architectures, we have used a standard stacked convolutional approach, with the model entitled "DistGP-DeepConv" consisting of 64 hidden units for the "Convolutionally Warped DistGP" part of the module, respectively, 5 hidden units for the "DistGP activation-function" part. For the DeepConvGP, we used 64 hidden units at each hidden layer. All models use a stride of 2 at the first layer. In all experiments we use 250 inducing points at each layer. Lastly, we also devised 18 hidden layers size versions of the ResNet (He et al. 2016) and DenseNet (Huang et al. 2017) architectures to see how our proposed module fares on complex architectures.

Table 4.1 shows the classification accuracy on MNIST and CIFAR-10 for different Convolutional GP



(a) Collapsed SGPR **(b)** DistGP-NN

Figure 4.4: Reliability of *in-between* uncertainty. Top row: Predictive mean and variance of parametric part of SGP; **Bottom row:** Predictive mean and variance of non-parametric part of SGP.



(a) Collapsed SGPR **(b)** DistGP-NN

Figure 4.5: Reliability of epistemic uncertainty. Top row: Predictive mean and variance of parametric part of SGP; **Bottom row:** Predictive mean and variance of non-parametric part of SGP.

models. Compared to other convolutional GP approaches, our method achieves superior classification accuracy compared to DeepConvGP (Blomqvist et al. 2018). We find that for our method, adding more layers increases the performance significantly. This observation is only available for a couple of stacked layers, as the results from our ResNet and DenseNet variants do not support this assertion. The GPDNN models introduced in Bradshaw et al. (2017) are nonetheless close to state of the art on CIFAR10 but also using a variant of DenseNet (Huang et al. 2017) as the building blocks for their GP classifier.

Convolutional GP models	Hidden Layers	MNIST	CIFAR-10
ConvGP	0	98.83	64.6
DeepConvGP	1	98.38	58.65
DeepConvGP	2	99.24	73.85
DeepConvGP	3	99.44	75.89
DistGP-DeepConv	1	99.01	70.12
DistGP-DeepConv	2	99.43	76.54
DistGP-DeepConv	3	99.67	78.49
DistGP-ResNet-18	18	99.52	74.56
DistGP-DenseNet	18	99.75	75.29
Hybrid NN-GP models	Hidden Layers	MNIST	CIFAR-10
Deep Kernel Learning	5	99.2	77.0
GPDNN	40	**99.95**	**93.0**

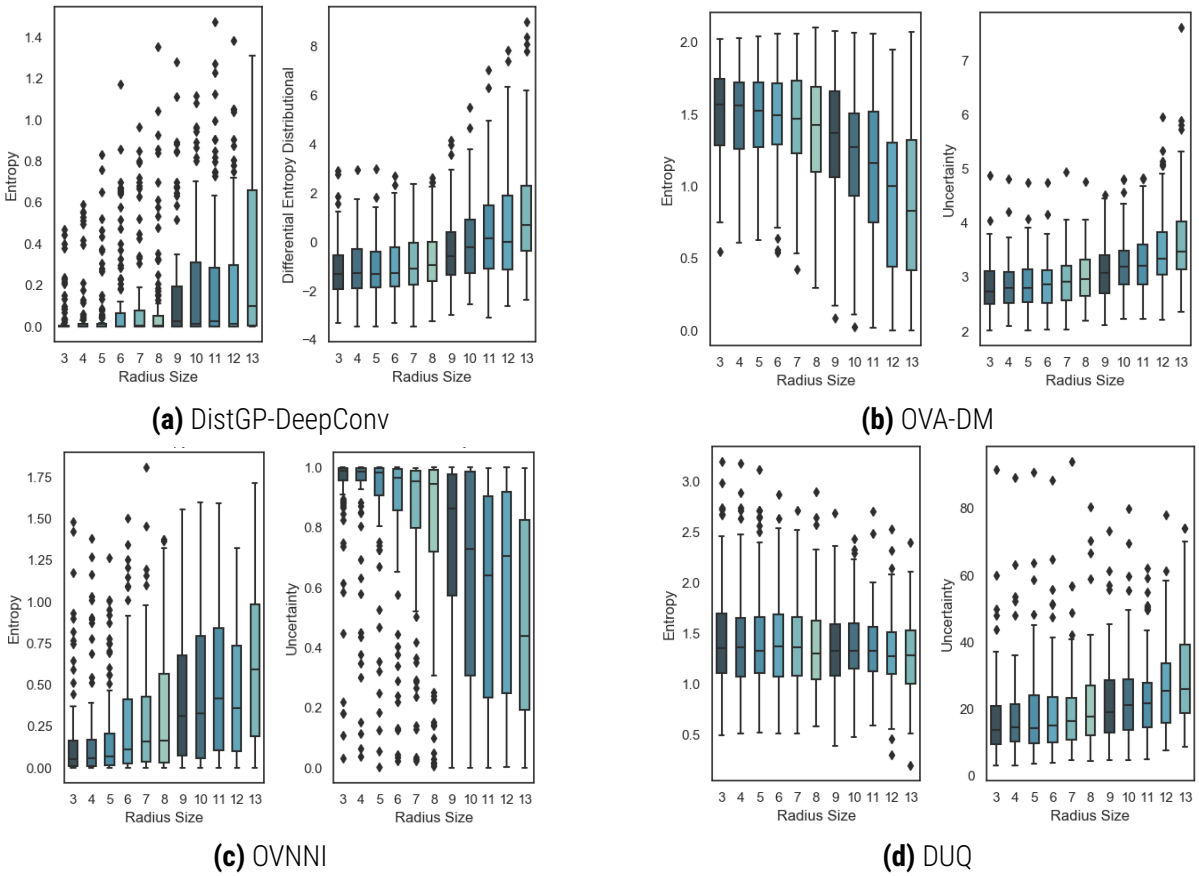
Table 4.1: Performance on MNIST and CIFAR-10. Deep Kernel Learning are the set of models from Wilson et al. (2016), whereas GPDNN are the set of models published in Bradshaw et al. (2017). Other results than our method are taken from the respective publications. Highlighted in bold are the best results.

Outlier detection on different fonts of digits. We test if DistGP-DeepConv models outperform OOD detection models from literature such as DUQ (Amersfoort et al. 2020), OVA-DM (Padhy et al. 2020) and OVVNI (Franchi et al. 2020). In these experiments we assess the capacity of our model to detect domain shift by training it on MNIST and looking at the uncertainty measures computed on the testing set of MNIST and the entire NotMNIST dataset (Bulatov 2011), respectively SVHN (Netzer et al. 2011). The hypothesis is that we ought to see both higher predictive entropy and differential entropy for distributional uncertainty (respectively higher OOD measures specific to each of the baseline models) for the digits stemming from a wide array of fonts present in NotMNIST as none of the fonts are handwritten, respectively the digit fonts in SVHN exhibit different backgrounds, orientations besides not being handwritten.

From Table 4.2 we can observe that generally all models exhibit a shift in their uncertainty measure between MNIST and NotMNIST, with the notable exception of OVNNI which barely manages to better separate the two datasets compared to a random guess. Moreover, OVA-DM manages to completely separate the two datasets with the caveat that it obtains lower predictive entropy for MNIST vs. NotMNIST compared to DistGP-DeepConv. The latter achieves similar results to DUQ, with the added benefit of a higher degree of separation using predictive entropy. In the case of SVHN we can observe similar patterns to NotMNIST, with OVA-DM and DistGP-DeepConv managing to almost separate the two datasets (MNIST vs. SVHN) by inspecting their uncertainty measure, again with the caveat for OVA-DM that it exhibits lower predictive entropy for SVHN in comparison to MNIST.

Model	MNIST vs. NotMNIST AUC		MNIST vs. SVHN AUC	
AUC	Pred. Entropy	OOD measure	Pred. Entropy	OOD measure
DistGP-DeepConv	**0.92**	0.82	**0.95**	0.98
0VA-DM	0.73	**1.0**	0.70	**1.0**
OVNNI	0.68	0.55	0.56	0.81
DUQ	0.82	0.81	0.65	0.74

Table 4.2: OOD detection results. Performance of OOD detection based on predictive entropy and distributional differential entropy (for baseline OOD models each has a different OOD measure). Models are trained on MNIST (normative data). Highlighted in bold are the best results.



(a) DistGP-DeepConv **(b)** OVA-DM

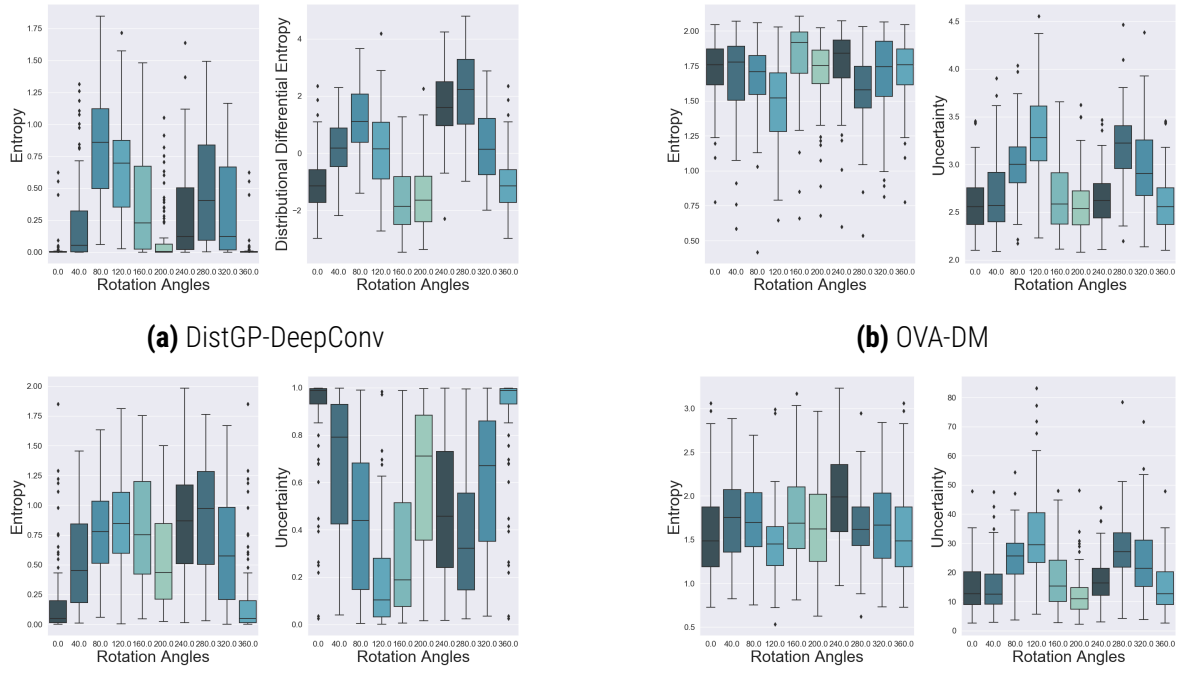
(c) OVNNI **(d)** DUQ

Figure 4.6: Sensitivity to morphological deformations. Predictive entropy and model-specific uncertainty measure for varying models as swelling of increasing radius is applied on MNIST digits. Higher values of uncertainty measure indicate outlier status, expect for OVNNI where the inverse is true. Results are shown for 3 hidden layers with DistGP-DeepConv dimensionality being set to 5, whereas the capacity of the convolutionally warped DistGP was set to 12, whereas for OOD models we use 128 hidden units at each layer.

Sensitivity to input perturbations. MorphoMNIST (Castro et al. 2018) enables the systematic deformation of MNIST digits using morphological operations. We use MorphoMNIST to better

understand the outlier detection capabilities of each method by exposing them to increasingly deformed samples. We use the first 500 MNIST digits in the testing set to generate new images with controlled morphological deformations. We use the swelling deformation with a strength of 3 and increasing radius from 3 to 14. Our hypothesis is that the predictive entropy should increase as the deformation is increased, alongside with the distributional differential entropy, which is a measure of the overall uncertainty in the logit space. This is motivated by the fact that the newly obtained images from MorphoMNIST are outside of the data manifold, which is different from the concept of having high uncertainty as expressed by entropy upon seeing a difficult digit to classify. In this case we would expect high entropy but low differential entropy.

All models are able to pick up on the shift in the data manifold as swelling is applied to the original digits, with the model-specific uncertainty measure steadily increasing (for OVNNI, a decrease in the measure translates to higher uncertainty) as increasing deformation is applied. However, for OVA-DM and DUQ the predictive entropy is stable or actually decreases as more deformation is applied, which is in contrast to what one would expect (Figure 4.6).



(a) DistGP-DeepConv **(b)** OVA-DM

(c) OVNNI **(d)** DUQ

Figure 4.7: Sensitivity to image rotation. Predictive entropy and model-specific uncertainty measure for varying models as varying degrees of rotation is applied to digit 6. Higher values of uncertainty measure indicate outlier status, expect for OVNNI where the inverse is true. Results are shown for 3 hidden layers with DistGP-DeepConv dimensionality being set to 5, whereas the capacity of the convolutionally warped DistGP was set to 12, whereas for OOD models we use 128 hidden units at each layer.

To further assess the sensitivity to input perturbations of our methods, we employ the experiments introduced in Gal and Ghahramani (2016b) by successively rotating digits from MNIST. We expect

to see an increase in both predictive entropy and distributional differential entropy as digits are rotated. For our experiment we rotate digit 6. When the digit is rotated by around 180 degrees the entropy and differential entropy should revert back closer to initial levels, as it will resembles digit 9.

From Figure 4.7 we can notice that all models exhibit an increase (decrease for OVNNI translates into higher uncertainty) in their specific uncertainty measures for rotation angles between 40 and 160, respectively between 240 and 320 degrees. In terms of predictive entropy, we can discern relatively stable and highly overlapping values for OVA-DM and DUQ, whereas for DistGP-DeepConv and OVNNI we can observe a clear pattern of increases and decreases as what was originally a 6 becomes a 9.

4.5 DistGP-based Segmentation Network & OOD Detection in Medical Imaging

As we have previously mentioned in section 4.1, the final aim of this chapter is to construct segmentation models for medical images that besides being accurate at segmenting different brain structures, can also offer reliable uncertainty quantification for their predictions and to disentangle the sources of uncertainty in such a way as to enable precise OOD detection.

DistGP-Seg architecture. The probabilistic module introduced in Sec. 4.3 can be used to construct a convolutional network that benefits from properties of DistGP. Specifically, we construct a 3D network for segmenting volumetric medical images, which is depicted in Figure 4.8. It consists of a convolved GP layer, followed by two measure-preserving DistGP layers. Each hidden layer uses filters of size $5\times5\times5$. To increase the model's receptive field, in the second layer we use convolution dilated by 2. We use 250 inducing points and 2 channels for the DistGP "activation functions". The affine operators project the stochastic patches into a 12 dimensional space. The size of the network is limited by computational requirements for GP-based layers, which is an active research area. Like regular convolutional nets, this model can process input of arbitrary size but GPU memory requirement increases with input size. We here provide input of size 32^3 to the model, which then segments the central 16^3 voxels. To segment a whole scan we divide it into tiles and stitch together the segmentations.

Evaluation on Brain MRI. In this section we evaluate our method alongside recently introduced OOD detection models (Amersfoort et al. 2020; Franchi et al. 2020; Padhy et al. 2020), assessing their capabilities to reach segmentation performance comparable to well-established deterministic models and whether they can accurately detect outliers.

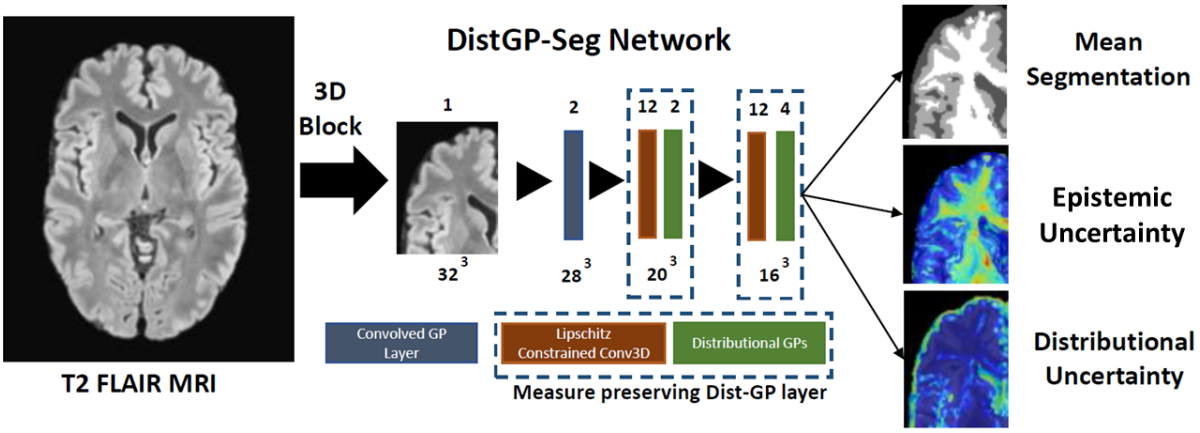


Figure 4.8: Schematic of proposed DistGP activated segmentation net. 3D blocks are extracted from MRI scan, being fed into DistGP-Seg network. Above and below each layer we show the number of channels and their respective dimensions. Each layer consists of Lipschitz constrained convolutional filters, followed by Distributional GP acting as "activation functions". Network offers voxel-level predictions of tissue class (*i.e.*, background, cerebrospinal fluid, white matter and grey matter). Using logit space predictive distributions we compute differential entropy for epistemic and distributional uncertainty to quantify uncertainties.

4.5.1 Data and pre-processing

For evaluation we use the following datasets:

1) Brain magnetic resonance imaging (MRI) scans from the UKBB study (Alfaro-Almagro et al. 2018), which contains scans from nearly 15,000 subjects. We selected for training and evaluation the scans which are in the bottom 10% percentile in terms of volume of white matter hyperintensities (see Figure B.10 for examples), with an equal split between training and testing. All subjects have been confirmed to be normal by radiological assessment. Segmentation of brain tissue (CSF, GM, WM) has been obtained with SPM12.

2) MRI scans of 285 patients with gliomas from BraTS 2017 (Bakas et al. 2017). All classes are fused into a *tumor* class, which we will use to quantify OOD detection performance.

In what follows, we use only the Fluid-attenuated inversion recovery (FLAIR) sequence (see Figure 4.9 for example) to perform the brain tissue segmentation task and OOD detection of tumors, as this MRI sequence is available for both UKBB and BraTS. FLAIR sequences reveal tissue T2 prolongation with cerebrospinal fluid suppression, allowing detection of superficial brain lesions (Bakshi et al. 2001). All FLAIR images are pre-processed with skull-stripping, N4 bias correction, rigid registration to MNI152 space and histogram matching between UKBB and BraTS. Finally, we normalize intensities of each scan via linear scaling of its minimum and maximum intensities to the [-1,1] range.

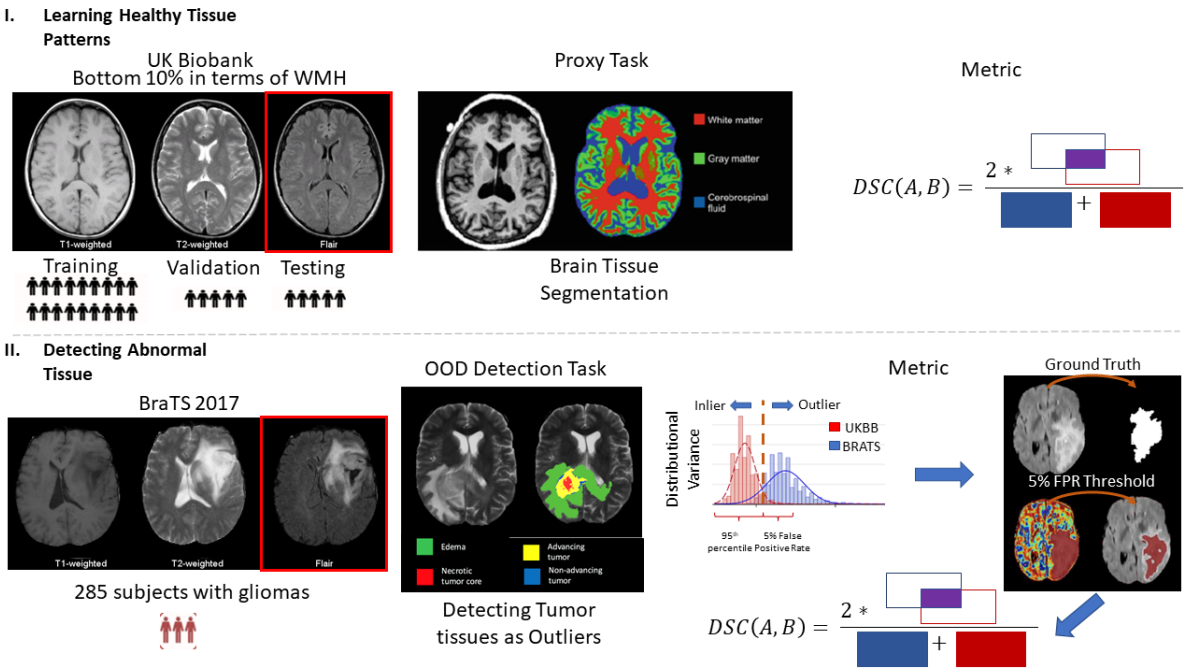


Figure 4.9: Schematic of study design. Top row: Segmenting brain tissue on healthy subjects serves both as a means to quantify the predictive capabilities of studied models but also as a proxy task for OOD detection as models are trained to segment healthy brain tissue; **Bottom row:** OOD detection capabilities are measured by how well the models can discern brain tumors as outliers based on a threshold on their uncertainty measure. All experiments use just FLAIR scans.

Model	Hidden Layers	DICE CSF	DICE GM	DICE WM
OVA-DM (Padhy et al. 2020)	3	0.72	0.79	0.77
OVNNI (Franchi et al. 2020)	3	0.66	0.77	0.73
DUQ (Amersfoort et al. 2020)	3	0.745	0.825	0.781
DistGP-Seg (ours)	3	0.829	0.823	**0.867**
U-Net	3 scales	**0.85**	**0.89**	0.86

Table 4.3: Dice scores on tissue segmentation. Performance on UK Biobank in terms of Dice scores per tissue. Higher Dice scores translate into improved segmentation capabilities. Highlighted in bold are the best results.

4.5.2 Brain tissue segmentation on normal MRI scans

Task: We train and test our model on the task of segmenting brain tissue of healthy UKBB subjects. This corresponds to the within-data manifold in our setup. The Dice score is commonly used to report segmentation results. We use the following formula $DSC = \frac{2|X \cap Y|}{|X|+|Y|}$, where $|X|$ and $|Y|$ represent the cardinalities of the two sets (*i.e.*, the number of elements in each set). In our case, we consider X to be the gold-standard segmentation of a given patch and Y to be the segmentation given by a machine learning model for the same patch. Higher values of the Dice score indicate a more efficient segmentation algorithm. A schematic of this experiment is given in

Figure 4.9.

Baselines: We compare our model with recent "OOD detection models" for enabling task-specific models (such as image segmentation) to perform uncertainty-based OOD detection (Amersfoort et al. 2020; Franchi et al. 2020; Padhy et al. 2020). For fair comparison, we use these methods in an architecture similar to ours (Figure 4.8), except that each layer is replaced by standard convolutional layer, each with 256 channels, LeakyRelu activations, and dilation rates as in ours. We also compare these Bayesian methods with a well-established deterministic baseline, a U-Net with 3 scales (down/up-sampling) and 2 convolution layers per scale in encoder and 2 in decoder (total 12 layers).

Results: Table 4.3 shows that DistGP-Seg surpasses other Bayesian methods with respect to Dice score for all tissue classes. Our method approaches the performance of the deterministic U-Net, which has a much larger architecture and receptive field. We emphasize this has not been previously achieved with GP-based architectures, as their size (*e.g.*, number of layers) is limited due to computational requirements. This supports the potential of DistGP, which is bound to be further unlocked by advances in scaling GP-based models.

4.5.3 Outlier detection in MRI scans with tumors

Model	DICE FPR=0.1	DICE FPR=0.5	DICE FPR=1.0	DICE FPR=5.0
OVA-DM (Padhy et al. 2020)	0.382	0.428	0.457	0.410
OVNNI (Franchi et al. 2020)	≤ 0.001	≤ 0.001	≤ 0.001	≤ 0.001
DUQ (Amersfoort et al. 2020)	0.068	0.121	0.169	0.182
DistGP-Seg (ours)	**0.512**	**0.571**	**0.532**	**0.489**
VAE-LG (Chen et al. 2019)	0.259	0.407	0.448	0.303
AAE-LG (Chen et al. 2019)	0.220	0.395	0.418	0.302

Table 4.4: Detecting brain tumors as outliers. Performance comparison of Dice for detecting outliers on BraTS for different thresholds obtained from UKBB. Higher Dice scores translate to increased OOD detection capabilities. Highlighted in bold are the best results.

Task: The previous task of brain tissue segmentation on UKBB serves as a proxy task for learning patterns of healthy tissues with our network. Here, we apply this pre-trained network on BraTS scans with tumors. We expect the region surrounding the tumor and other related pathologies, such as squeezed brain parts or shifted ventricles, to be highlighted with higher distributional uncertainty (or model-specific OOD measures from our range of "OOD detection models"), which is the

OOD measure for our proposed model. To evaluate quality of OOD detection at a pixel level, we follow the procedure in Chen et al. (2019), for example to get the 5.0% False Positive Ratio threshold value we compute the 95% percentile of distributional variance on the testing set of UKBB, taking into consideration that there is no outlier tissue there. Subsequently, using this value we threshold the distributional variance heatmaps on BraTS, with tissue having a value above the threshold being flagged as an outlier. We then quantify the overlap of the pixels detected as outliers (over the threshold) with the ground-truth tumor labels by computing the Dice score between them. A schematic of this experiment is given in Figure 4.9.

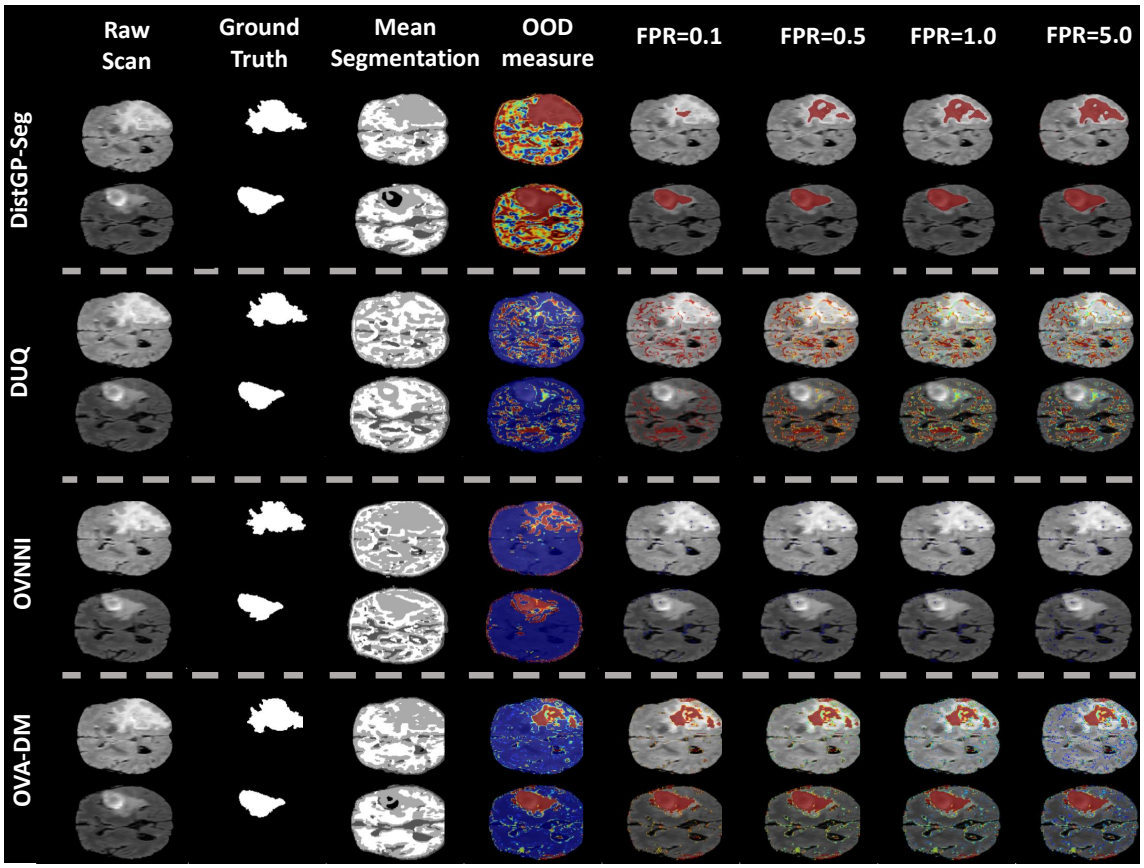
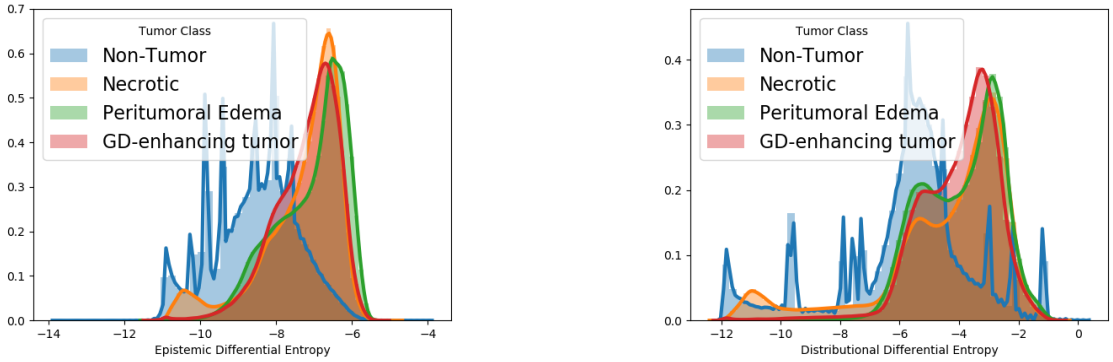


Figure 4.10: Case studies of brain tumor OOD detection. Comparison between models in terms of voxel-level outlier detection of tumors on BraTS scans. Mean segmentation represents the hard segmentation of brain tissues. OOD measure is the quantification of uncertainty for each model, using their own procedure. Higher values (towards the red spectrum) translate to the respective tissue being detected as an outlier, whereas for OVNNI it is the opposite (values towards the dark blue spectrum). OOD measures have been normalized to be between 0 and 1 for each model in part, based on their own OOD measure procedure.

Results: Table 4.4 shows the results from our experiments with DistGP and compared "OOD detection models". We also provide performance of reconstruction-based OOD detection models as reported in Chen et al. (2019) for similar experimental setup. DistGP-Seg surpasses its counterparts, as well as reconstructed-based models. In Figure 4.10 we provide representative results from the methods we implemented for qualitative assessment. DistGP-Seg provides smooth estimates

in the OOD measure column due to its Lipschitz constrained layers and the inherent smoothness present in its RBF kernels. Due to these reasons, the OOD measure column is less sharp compared to other ones. Conversely, even in the case of DUQ, which is Lipschitz constrained, the resulting OOD measures from it are sharp. Perhaps surprising, OVNNI does seem to point out the outlier region with varying degrees of success, however in the wrong direction. Concretely, it seems to highlight the tumor region as an inlier. Moreover, although BraTS does not provide labels for WM/GM/CSF tissues hence we cannot quantify how well these tissues are segmented, visual assessment shows our method compares favorably to compared counterparts.

In Figure 4.11 we plotted the different differential entropy measures based on BraTS scans by overlying their tumor labels on the obtained uncertainties from our model. We can notice that tumor tissue is highlighted with higher epistemic and distributional uncertainty compared to healthy tissue. More detailed plots are available in Appendix B.2.1.



(a) Epistemic Differential Entropy

(b) Distributional Differential Entropy

Figure 4.11: Uncertainty on brain tumor tissue classification. Comparison in terms of voxel-level epistemic and distributional differential entropy between non-tumor tissues and different tumor gradations (see Figure 4.9 for details regarding tissues) from subjects in the BraTS dataset using DistGP-Seg. Higher differential entropy values translate into more uncertainty with regards to predictions.

4.6 Discussion

We have introduced a novel Bayesian convolutional layer with Lipschitz continuity that is capable of reliably propagating uncertainty. We have shown on a wide array of general OOD detection tasks that it surpasses other OOD models from literature, while also offering an increase in accuracy compared to counterpart architectures based solely on Euclidean space SVGP (Blomqvist et al. 2018). General criticism surrounding deep and convolutional GP involves the issue of underperformance compared to other Bayesian deep learning techniques, and especially compared to deterministic networks. Our experiments demonstrate that our 3-layers model, size limited due to computational cost, is capable of approaching the performance of a U-Net, an architecture with

a much larger receptive field. Further advances in computational efficient GP-based models, an active area of research, will enable our model to scale further and unlock its full potential. Importantly, we showed that our DistGP-Seg network offers better uncertainty estimates for OOD detection than state-of-the-art OOD detection models, and also surpasses some recent unsupervised reconstruction-based deep learning models for identifying outliers corresponding to pathology on brain scans.

This framework can also be used for regression and classification tasks within a medical imaging context, facilitating the adoption of deep learning in clinical settings thanks to enhanced accountability in predictions. For example, parts of scans flagged with high distributional uncertainty can be sent back for inspection and quality control. To support our claim, we have included additional results on flagging white matter hyperintensities as outliers (see Appendix B.2.2), respectively retina pathologies (see Appendix B.2.3).

Our results indicate that OOD methods that do not take into account distances in latent space, such as OVNNI, tend to fail in detecting outliers, whereas OVA-DM and DUQ that make predictions based on distances in the last layer perform better. Our model utilises distances at every hidden layer, thus allowing the notion of outlier to evolve gradually through the depth of our network. This difference can be noticed in the smoothness of OOD measure for our model in comparison to other methods in Figure 4.10. Furthermore, the issue of *feature collapse* (Amersfoort et al. 2020) in deep networks can be precisely controlled due to the mathematical underpinnings of our proposed network, enabling us to assess the scenarios when this happens. Additionally, we have shown that despite the possibility of achieving over-correlation in the latent space via the loss function, that this does not happen in practice for our proposed module regardless of whether we have imposed Lipschitz continuity. Perhaps surprisingly, our DKL model trained on the "snelson" dataset managed to achieve a smoothness in predictive distributions, respectively a "locality" of correlations in its last hidden layer on par with collapsed SGPR. We emphasize that this this behaviour was not seen in other works (Ober et al. 2021) and could be influenced by network architecture and different training schedule.

Concurrent work (Morales-Alvarez et al. 2020) introduced a similar module, whereby they replace activation functions with GP with triangular kernels that are designed to imitate the behaviour of the ReLu non-linearity. The authors do not report OOD detection experiments.

A drawback of our study resides in the small architecture used on medical imaging scans. Extending our "measure preserving DistGP" module to larger architectures such as U-Net for segmentation or modern CNN for whole-image prediction tasks remains a prospective research avenue fuelled by advances in scalability of SVGP (in the next chapter we shall introduce such a method to eliminate computationally expensive matix operations in SVGP). Moreover, our experiments involving more complicated architectures, such as ResNet or DenseNet for standard multi-class classification, have not managed to surpass in accuracy a far less complex model with only 3 hidden layers.

A plausible reason behind this under-fitting resides in the factorized approximate posterior formulation, which was shown to negatively affect predictive performance compared to MCMC inference schemes (Havasi et al. 2018). We posit that using alternative inference frameworks (Ustyuzhaninov et al. 2019) whereby we impose correlations between layers might alleviate this issue. Moreover, the lack of added representational capacity upon adding new layers raises some further questions regarding what are optimal architectures for hierarchical GP, what inductive biases do they need or how to properly initialize them to facilitate adequate training. Additionally, our comparison with respect to reconstruction based approaches towards OOD detection was not complete as it did not include a comprehensive list of recent models (Baur et al. 2018; Dey and Hong 2021; Pinaya et al. 2021; Schlegl et al. 2019). However, comparing our proposed model with reconstruction based approaches was not our intended goal for this chapter, the main aim being to compare with models which can provide accurate predictive results alongside OOD detection capabilities at the same time.

In conclusion, our work shows that incorporating DistGP in convolutional architectures provides both competitive performance and reliable uncertainty quantification in medical image analysis alongside general OOD tasks, opening up a new research direction.

Chapter 5

Matrix inversion-free inference in Student-t Processes

This chapter is based on the following publication:

(a) **Popescu, S. G.**, Glocker, B., and Wilk, M. Van der (2022). "Matrix Inversion free variational inference in Conditional Student's T Processes". In: *Symposium on Advances in Approximate Bayesian Inference* – (Popescu et al. 2022)

The code for all experiments is available at https://github.com/SebastianPopescu/InverseFreeStudent-TProcesses.

We propose a new variational lower bound for performing inference in sparse Student-t Processes that does not require computationally intensive operations such as matrix inversions or log determinants of matrices. We devise a mathematically valid and easy-to-sample from approximate posterior over the Inverse Wishart distributed covariance matrix encompassing both training data and inducing points. To deal with remaining log determinant terms we propose a conjugate gradient based lower bound that is tight when approximate posteriors are optimal. We demonstrate that our proposed model behaves similarly to matrix inversion dependent counterparts in terms of convergence of evidence lower bound and predictive capabilities at testing time on a wide array of toy experiments. Moreover, we test the validity of our method on medium and large scale datasets, showing encouraging results.

5.1 Introduction

The major drawback of training GP resides in the computational complexity of inverting the covariance matrix, which is cubic in computation and quadratic in memory. This has motivated research on sparsifying GP (Quinonero-Candela and Rasmussen 2005; Seeger et al. 2003). Titsias (2009) has addressed this problem by leaving the prior distribution of the GP unchanged, with sparsity being enforced in the posterior through inducing points learnt by variational inference. Hensman et al. (2013) proposed an inducing point framework scalable to large data, obtaining posterior formulas conditioned on these artificial points. However, this scales supralinearly with regards to inducing point numbers, resulting in $\mathcal{O}(M^2N + M^3)$ computation and $\mathcal{O}(MN + M^2)$ storage complexity, where M is the number of inducing points. Detailed discussion on aforementioned attempts to sparsify GP is given in subsection 2.2. Recent work has focused on reducing the computational cost associated to large inducing point numbers by imposing grid structures on their locations to exploit faster matrix algebra routines (Evans and Nair 2018; Gardner et al. 2018a; Wilson and Nickisch 2015). Therefore, a major obstacle towards the wider adoption of GP in large scale datasets is related to the computational cost of matrix inversion and log determinants. With this in mind, in Wilk et al. (2020b) the authors propose a lower bound that can be computed without computationally expensive matrix operations such as inversion or log determinants (details are provided in subsection 2.3.2). Similar in scope, we propose to learn variational approximations of the covariance matrix, implicitly also over inverses, of inducing and training points, thereby proposing a computationally efficient approximate posterior over covariance matrices within the probabilistic framework of Student-t Processes (TP) (Shah et al. 2014). Using properties of Inverse Wishart distributions, we can obtain easy-to-sample from distributions over the random variables $\Sigma_{fu}\Sigma_{uu}^{-1}$ and $\Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}$, thus eliminating inversion operations in the predictive equations both at training and testing time, respectively. Compared to Wilk et al. (2020b), where the authors solve the issue of inverse-free predictive equations by using a highly structured posterior $q\left(U\right)$, we obtain similar properties by virtue of our Bayesian hierarchical process, with an additional KL divergence term over our approximation of K_{uu}^{-1}, thus favouring the retrieval of the exact solution. Lastly, the motivation behind this chapter is in devising inverse-free GP so as to diminish the long training time of models introduced in Chapter 4.

Relation between Student-t Processes and outliers. In previous chapters of this thesis, we have focused solely on GP, so the motivation and rationale behind the usage of TP in this chapter is warranted at this moment. Besides the mathematical underpinnings of TP, which involve assigning a prior distribution over covariance matrices, thus facilitating the imposition of approximate posteriors over various matrices present in the resulting ELBO or predictive equations, there is also another rationale behind using TP over GP, namely that the former are more prone to avoid fitting outliers in the data. This phenomenon has been long known in the GP community, with early works

such as Neal (1997) using the Student-t distribution as observation noise in the context of a GP regression task. Whereas Neal (1997) used MCMC to perform inference, subsequent papers have reduced the computational cost by using approximate inference methods such as VI (Tipping and Lawrence 2005), expectation propagation (Jylänki et al. 2011) or Laplace approximation (Vanhatalo et al. 2009). Tipping and Lawrence (2005) provides an argument for the usage of TP from a robustness to outliers perspective. They make the case that if observations are contaminated by outliers (*e.g.*, corrupted observations), the accuracy of the predictive distribution can be severely affected if using a Gaussian noise assumption (see Figure 5 in Tipping and Lawrence (2005) for an illustrative example in case of corrupted observations). With these observations in mind and taking into consideration our intended goal of devising probabilistic models that can discern between inliers and outliers, the fact that TP avoid fitting corrupted observations or other types of outliers that might be present in the training data, should theoretically lead to enhanced OOD detection capabilities.

Contributions In this chapter we make the following contributions:

- propose a sparse variational framework for Student-t Processes.
- introduce a matrix inversion free variational lower bound for sparse Student-t Processes that converges to matrix inversion dependent bounds on a range of toy datasets.
- devise an approximate posterior over partitioned matrices $\Sigma_{fu,fu}$ that is both sample efficient and guarantees positive semi-definite samples.
- propose an iterative lower bound on matrix log determinants that converges in one iteration if preconditioner is optimal.
- show that Natural Gradient update rules avert the necessity of iterative methods, obtaining an analytic formula for optimizing variational parameters.

5.2 Background

In this section we introduce the necessary theoretical background needed for this chapter. A conceptual difference between TP and GP is that the former's covariances have priors, thus being able to handle uncertainties stemming from choosing a covariance function, a property that GP lack. Thus, we commence by introducing the mathematical basics of two distributions on positive definite matrices, the Wishart and Inverse Wishart distributions (Dawid 1981). Subsequently, we provide a brief summary of current frameworks for training and performing predictions with TP, which will later be extended to the sparse case in a similar fashion to the SVGP framework introduced in subsection 2.2.2. As we shall later see in this chapter, performing approximate inference in our proposed inverse-free sparse TP can be easier from a computational perspective if we use

natural gradients instead. Hence, this section will end with a comprehensive introduction to natural gradients.

5.2.1 Distributions on positive definite matrices

Wishart distributions are probability distributions defined on real-valued positive-definite matrices. Hence, we have $X \sim \mathcal{W}_m(\nu, \Psi)$ with Ψ being an $m \times m$ positive definite matrix and $\nu > m-1$ representing the degrees of freedom. The p.d.f. of this distribution is defined as $\frac{|x|^{\frac{\nu-m-1}{2}} \exp -\operatorname{Tr}(\Psi^{-1}x)/2}{2^{\frac{\nu m}{2}}|\Psi|^{\nu/2}\Gamma_m(\frac{\nu}{2})}$, where Γ_m denotes the multivariate gamma function. The mean of the Wishart distribution is given by $\mathbb{E}(X) = \nu\Psi$ and the variance is $\mathbb{V}(X_{ij}) = \nu\left(\Psi_{ij}^2 + \Psi_{ii}\Psi_{jj}\right)$.

Sampling Wishart Distributions. For this operation we use the Bartlett decomposition of the matrix $X \sim \mathcal{W}_p(\nu,\ \Sigma)$, which can be written in the following factorized manner $X = LAA^\top L^\top$, where A has the following matrix form $A = \begin{pmatrix} c_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ n_{p1} & \dots & c_p \end{pmatrix}$, where $n_{ij} \sim \mathcal{N}(0,1)$ and $c_i^2 \sim \chi^2_{\nu-i+1}$ is the chi-squared distribution and $LL^\top = \Sigma$.

Perhaps a natural questions which might arise now is whether we can place a prior on deterministic operations on such matrices, such as for example matrix inversion. We now detail this exact distribution.

Inverse-Wishart distributions are probability distributions defined on real-valued positive-definite matrices. We have $X \sim \mathcal{W}_m^{-1}(\nu, \Psi)$ with Ψ being an $m \times m$ positive definite matrix and $\nu > m-1$ representing the degrees of freedom. The p.d.f. of this distribution is defined as follows $\frac{|\psi|^{\nu/2}}{2^{\nu m/2}\Gamma_m(\frac{\nu}{2})}|x|^{-(\nu+m+1)/2}\exp -\frac{1}{2}\operatorname{Tr}(\Psi X^{-1})$. The mean of the inverse-Wishart is given by $\mathbb{E}(X)\frac{\Psi}{\nu-m-1}$ for $\nu \geq p+1$. The variance of individual elements is given as follows $\mathbb{V}(X_{ij}) = \frac{(\nu-p+1)\Psi_{ij}^2+(\nu-p-1)\Psi_{ii}\Psi_{jj}}{(\nu-p)(\nu-p-1)^2(\nu-p-3)}$.

Sampling Partitioned Inverse-Wishart Distributions From Theorem 3.2.10 in Muirhead (1982) we know that considering $A \sim \mathcal{W}_p(\nu, \Sigma)$ and for sampled partitioned matrices and mean partitioned covariance matrices of the following form:

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix};\ \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}, \tag{5.1}$$

where $A_{11} \in M_{m\times m}$ and $\Sigma_{11} \in M_{m\times m}$, we have the following properties:

$$A_{22} \sim \mathcal{W}_{p-m}\left(\nu,\, \Sigma_{22}\right) \tag{5.2}$$

$$A_{11\cdot 2} \sim \mathcal{W}_{m}\left(\nu - p + m,\, \Sigma_{11\cdot 2}\right) \tag{5.3}$$

$$A_{12}A_{22}^{-1} \mid A_{11\cdot 2} \sim \mathcal{MN}\left(\Sigma_{11}\Sigma_{22}^{-1},\, A_{11\cdot 2} \otimes \Sigma_{22}^{-1}\right)\,, \tag{5.4}$$

where $\mathcal{MN}$ is the matrix normal and $\otimes$ is the Kronecker product. Hence, if we would like to sample from the Inverse Wishart distributed partitioned matrix:

$$R = A^{-1} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}^{-1} = \begin{pmatrix} A_{11\cdot 2}^{-1} & -A_{11\cdot 2}^{-1}A_{12}A_{22}^{-1} \\ -A_{22}^{-1}A_{21}A_{11\cdot 2}^{-1} & A_{22}^{-1} + A_{22}^{-1}A_{21}A_{11\cdot 2}^{-1}A_{12}A_{22}^{-1} \end{pmatrix}\,, \tag{5.5}$$

we can see that we can easily sample from the Matrix Variate Normal distributed $A_{12}A_{22}^{-1}$. We still need to take samples from $A_{11\cdot 2}^{-1}$ and A_{22}^{-1}. Since we know that $A_{22} \sim \mathcal{W}_{p-m}\left(\nu, \Sigma_{22}\right)$, we can obtain samples from it using the Bartlett decomposition $A_{22} \sim L_{A_{22}}AA^{\top}L_{A_{22}}^{\top}$. Therefore, we now only need to invert $L_{A_{22}}A$. which is a lower triangular matrix. Same process is applied to obtain sample from $A_{11\cdot 2}^{-1}$. Needless to say, this method of sampling from partitioned Inverse Wishart distributions is only efficient if one already has access to $L_{A_{22}}$ and $L_{A_{11\cdot 2}}$. For a more general sampling procedure we refer the reader to section 4.5 of Shah et al. (2014).

Remark 5.2.1. *Intuitively since matrix inversion is a deterministic operation, $\mathcal{W}_p$ and $\mathcal{W}_p^{-1}$ are related. One can easily verify by doing necessary substitutions in their p.d.f. that if we have $X \sim \mathcal{W}_m\left(\nu, \Psi\right)$ then denoting by $A = X^{-1}$, we can say that $A \sim \mathcal{W}_m^{-1}\left(\nu, \Psi^{-1}\right)$.*

5.2.2 Student-t Processes

In this subsection, we detail the mathematical framework behind TP, how to train such models and perform predictions, alongside highlighting differences with GP.

A natural generalization of GP are the TP (Shah et al. 2014), which effectively take into consideration the uncertainty pertaining to choosing a covariance function, by placing a prior on them. A direct consequence of this is that TP exhibit heavier-tails compared to GP, thus making them more robust to outliers in the data (Solin and Särkkä 2015).

TP are stochastic processes that have the defining property that any finite collection of samples from them is distributed according to a multivariate Student-t distribution, which has the following density (Kotz and Nadarajah 2004):

$$p\left(f\right) = \frac{\Gamma\left(\frac{\nu+p}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\left(\nu\pi\right)^{\frac{p}{2}}\left|\Sigma\right|^{\frac{1}{2}}}\left(1 + \frac{1}{\nu}\left(f-\mu\right)^{\top}\Sigma^{-1}\left(f-\mu\right)\right)^{-\frac{\nu+p}{2}}\,, \tag{5.6}$$

where ν represents the degrees of freedom, p is the number of dimensions, Σ is the scaling matrix and ν is the location.

Remark 5.2.2. *Due to finite samples being distributed according to a multivariate Student-t distribution, more probability mass gets placed in the tails of the marginal distribution. This has a direct effect on the mean function, which shifts less to explain outliers (e.g., see Figure 5 in Neal (1997) for difference in behaviour of mean function between Gaussian and Student-t assumed observation noise).*

Generative process of Student-t Processes. *We follow the hierarchical Bayesian construction in prior space introduced in Shah et al. (2014):*

$$\Sigma_{ff} \sim \mathcal{W}_n^{-1}\left(\nu_p + n + 1, \nu_p K_{ff}\right) \tag{5.7}$$

$$f \mid \Sigma_{ff} \sim \mathcal{N}\left(0, \Sigma_{ff}\right) \tag{5.8}$$

In Shah et al. (2014), the authors provide a short derivation showing that by marginalizing out Σ_{ff} one obtains a multivariate t-distribution. We provide here a more detailed derivation taking into consideration that we have the general result:

$$\int p(\mathbf{X} \mid 0,\ \Sigma) p(\Sigma \mid \nu,\ \Psi) \mathrm{d}\,\Sigma = \frac{|\Psi|^{\nu/2} \Gamma_p(\frac{\nu+n}{2})}{\pi^{np/2} |\Psi + A|^{\frac{\nu+n}{2}} \Gamma_p(\frac{\nu}{2})}\,, \tag{5.9}$$

where $A = XX^\top$ and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ are independent p-variate Gaussian variables drawn from a $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ distribution. Adapting this result to our scenario we get that:

$$\frac{|\nu K_{ff}|^{\frac{\nu}{2}}\, \Gamma_n(\frac{\nu+n}{2})}{\pi^{n^2/2}\, |\nu K_{ff} + XX^\top|^{\frac{\nu+n}{2}}\, \Gamma_n(\frac{\nu}{2})}\,. \tag{5.10}$$

Focusing on the determinant terms we can rearrange:

$$\frac{|\nu K_{ff}|^{\frac{\nu}{2}}}{|\nu K_{ff} + XX^T|^{\frac{\nu+n}{2}}} = \frac{|\nu K_{ff}|^{-\frac{n}{2}}}{|I_n + \frac{1}{\nu} K_{ff}^{-1} XX^\top|^{\frac{\nu+n}{2}}}\,. \tag{5.11}$$

We remind the matrix determinant lemma : $\left|A + uv^\top\right| = \left(1 + v^T A^{-1} u\right)|A|$. We apply it to the denominator of the previous equation:

$$\left| I_n + \frac{1}{\nu} K_{ff}^{-1} XX^\top \right|^{\frac{\nu+n}{2}} = \left[1 + \frac{1}{\nu} X K_{ff}^{-1} X^\top \right]^{\frac{\nu+n}{2}}\,. \tag{5.12}$$

Plugging this into equation 5.10 we get that:

$$\frac{\Gamma_n(\frac{\nu+n}{2})}{\pi^{n^2/2}\Gamma_n(\frac{\nu}{2})\nu^{\frac{n^2}{2}}\left|K_{ff}\right|^{\frac{n}{2}}}\left[1+\frac{1}{\nu}X^\top K_{ff}^{-1}X\right]^{-\frac{\nu+n}{2}}, \tag{5.13}$$

which we can now recognize as a $\mathcal{MVT}(\nu,\,0,\,K_{ff})$, with $\mathcal{MVT}$ standing for multivariate t-distribution, which is a generalization to random vectors of the Student-t distribution. The p.d.f. of this distribution is defined as $\frac{\Gamma(\frac{\nu+n}{2})}{\Gamma(\frac{\nu}{2})\nu^{n/2}\pi^{n/2}|\Sigma|^{1/2}}\left[1+\frac{1}{\nu}(x-\mu)^\top\Sigma^{-1}(x-\mu)\right]^{-\frac{\nu+n}{2}}$, where μ is the location, Σ is a positive-definite $n*n$ matrix and ν represent the degrees of freedom. For $\nu\geq 1$ the mean is μ and for $\nu\geq 2$ the variance is defined as $\frac{\nu}{\nu-2}\Sigma$. Therefore, by starting from an Inverse-Wishart distributed prior over the covariance of a GP evaluated at X, we have implicitly imposed a Student-t distribution over our latent function f.

We can now proceed to define a Student-t process f. For any finite collection of function values we have a joint multivariate Student-t distribution $\{f_i\}_{i=1}^n\sim\mathcal{MVT}(\nu,\,0,\,K_{ff})$, or more succinctly $f\sim\mathcal{TP}(\nu,\,0,\,K)$, where we considered a zero mean function, ν represents the degrees of freedom and K is the chosen kernel. We know from subsection 2.1 that GP exhibit closed from solutions in the case of regression (i.e., marginal likelihood and conditional distribution). This is however not the case for TP if we assume a Student-t noise distribution, which is caused by the fact that Student-t distribution are not closed under addition (Kotz and Nadarajah 2004). Shah et al. (2014) averts this problem by incorporating the additive noise term in the kernel directly.

Predictions are obtained by making use of the conditional predictive distribution associated to multivariate Student-t distributions, which is given by the following formula (Ding 2016):

$$y\mid x\sim\mathcal{MVT}\left(\nu+n_x,\;\mu_y+K_{yx}K_{xx}\left(x-\mu_x\right),\;\beta\left(K_{yy}-K_{yx}K_{xx}^{-1}K_{xy}\right)\right), \tag{5.14}$$

where $\beta=\frac{\nu+(x-\mu_x)^\top K_{xx}^{-1}(x-\mu_x)}{\nu+n_x}$ and the subscripts indicate from which random variable they belong.

Remark 5.2.3. *One can remark that in the formula for the covariance of the conditional distribution, its values depend directly on the observed data x. Shah et al. (2014) makes the case that this feature leads to improved predictive covariances in their experiments comparing GP to TP.*

5.2.3 Natural Gradients

In this subsection we introduce all the required theoretical toolkit for understanding natural gradients, which will play a key role in subsequent parts of this chapter. A plethora of familiar distributions can be expressed in the following way:

$$q_\eta(\theta)=h(\theta)\exp\left[\langle\eta,\phi(\theta)\rangle-A(\eta)\right], \tag{5.15}$$

where $h(\theta)$ represents the base measure, η denotes the natural parameters, ϕ the sufficient statistics and $A(\eta)$ the log partition function.

Natural gradients take advantage of geometrical information pertaining to the distribution we are optimizing over, with the update taking the following form:

$$\eta_{t+1} = \eta_t + \beta_t F_{\eta_t}^{-1} \nabla_\eta L(\eta_t) \ , \tag{5.16}$$

where the Fisher Information Matrix (FIM) is given by $F_{\eta_t} = \mathbb{E}_{q_\eta(\theta)} \left[\nabla_\eta \log q_\eta(\theta) \nabla_\eta \log q_\eta(\theta)^T \right]$.

The minimal exponential family representation implies a one-to-one correspondence between the natural parameters η and mean parameters m, respectively $m = \mathbb{E}_{q_\eta(\theta)}[\phi(\theta)]$ and $m = \nabla_\eta A(\eta)$. To show the last statement we can see that:

$$\nabla_\eta \log q_\eta(\theta) = \phi(\theta) - \nabla_\eta A(\eta) \tag{5.17}$$

and taking an expectation over it results in:

$$\mathbb{E}_{q(\theta)} \nabla_\eta \log q_\eta(\theta) = \mathbb{E}_{q(\theta)} \phi(\theta) - \nabla_\eta A(\eta) \tag{5.18}$$

$$= m - \nabla_\eta A(\eta) \tag{5.19}$$

and since the score has a null expectation then we have that $m = \nabla_\eta A(\eta)$.

To avert having to invert the Fisher Information Matrix, we can make use of the following trick:

$$\nabla_\eta L(\eta_t) = \nabla_m L(m_t) \nabla_\eta m_t = \nabla_m L(m_t) \nabla^2_{\eta\eta} A(\eta_t) \ . \tag{5.20}$$

Using the fact that $\nabla^2_{\eta\eta} A(\eta_t) = F(\eta_t)$ and replacing them in the natural gradient update equation we get:

$$\eta_{t+1} = \eta_t + \beta_t \nabla_m L(m_t) \ . \tag{5.21}$$

Natural Gradient update rule with respect to mean parameters. *Taking into consideration the standard objective of a Variational Inference problem, where we need to maximize the ELBO:*

$$L(\eta) = \mathbb{E}_{q_\eta(\theta)}[\log p(D \mid \theta)] + \mathbb{E}_{q_\eta(\theta)} \left[\log \frac{p_{\eta_0}(\theta)}{q_\eta(\theta)} \right] \ . \tag{5.22}$$

We now focus on computing the derivatives with respect to the mean parameters of the KL-divergence

term:

$$\nabla_m \mathbb{E}_{q_\eta(\theta)}\left[\log \frac{p_{\eta_0}(\theta)}{q_\eta(\theta)}\right] = \nabla_m \mathbb{E}_{q_\eta}\left[\phi(\theta)^T\left[\eta_0 - \eta\right] + A(\eta) + c\right] \tag{5.23}$$

$$= \nabla_m\left[m^T\left[\eta_0 - \eta\right]\right] + \nabla_m A(\eta) \tag{5.24}$$

$$= \eta_0 - \eta - \nabla_m \eta^T m + \nabla_m A(\eta) \tag{5.25}$$

$$= \eta_0 - \eta - F(\eta)^{-1} m + F(\eta)^{-1} m \tag{5.26}$$

$$= \eta_0 - \eta\,, \tag{5.27}$$

where in the derivation we used the general equivalence $\nabla_m(\cdot) = F(\eta)^{-1}\nabla_\eta(\cdot)$.

We now have a much more simplified version of the natural gradient update rule:

$$\eta_{t+1} = \eta_t + \beta_t \nabla_m \mathbb{E}_{q_\eta(\theta)}\left[\log p(D \mid \theta)\right] + \beta_t\left[\eta_0 - \eta\right]\ . \tag{5.28}$$

The computation of the derivatives with respect to m of the expected data fit term will depend on the distribution we are optimizing over.

Natural Gradient update rule with respect to reparametrized natural parameters. *There might be instances when updating the natural parameters is not the optimal pathway for a certain problem, such as is the case with updating a p.s.d. matrix. The eventually derived update rule for Σ will not guarantee its positive semi-definitiveness. We make use of a general purpose framework introduced in Salimbeni et al. (2018). We denote by $\xi = \xi(\eta)$ the operation which transforms the natural parameters η into re-parametrized space ξ.*

An important result, which will prove useful later in the derivation, is that $\nabla^2_\eta \log(q(\theta)) = -\frac{\partial m}{\partial \eta}$.

Before we continue with the derivation, it is good to emphasize that we are interested in obtaining a natural gradient update rule for ξ, namely:

$$\xi_{t+1} = \xi_t - \beta_t F^{-1}_{\xi_t^T} \nabla_{\xi^T} L\left(\xi_t\right)\ , \tag{5.29}$$

where $F_\xi = -\mathbb{E}_{q(\theta;\xi)}\nabla^2_\xi \log q(\theta;\xi)$. Computing the FIM for natural parameters η is given by:

$$F_\eta = \frac{\partial m}{\partial \eta}\,, \tag{5.30}$$

which when inserted in the natural gradient update rule for natural parameters η gives the following

simplification:

$$\eta_{t+1} = \eta_t + \beta_t F_{\eta_t}^{-1} \nabla_\eta L\left(\eta_t\right) \tag{5.31}$$

$$= \eta_t + \beta_t \left(\frac{\partial m}{\partial \eta}\right)^{-1} \nabla_\eta L\left(\eta_t\right) \tag{5.32}$$

$$= \eta_t + \beta_t \nabla_m L\left(m_t\right) \ , \tag{5.33}$$

which gives us the update rule introduced in the previous subsection, albeit from a slightly different derivation. Not having to compute the inverse of FIM is essential, this being a feature we also want to imbue towards our ξ natural gradient update rule.

Using the chain rule, we can express the FIM_ξin terms of the FIM_η as follows:

$$FIM_\xi = \left[\frac{\partial \eta}{\partial \xi}\right]^T \frac{\partial m}{\partial \eta} \frac{\partial \eta}{\partial \xi} \ . \tag{5.34}$$

We remind ourselves of the important fact that the inverse of a Jacobian is the Jacobian of the inverse. With this in mind, we proceed:

$$\frac{\partial L}{\partial \xi} FIM_\xi^{-1} = \frac{\partial L}{\partial \xi} \left[\left[\frac{\partial \eta}{\partial \xi}\right]^T \frac{\partial m}{\partial \eta} \frac{\partial \eta}{\partial \xi}\right]^{-1} \tag{5.35}$$

$$= \frac{\partial L}{\partial \xi} \frac{\partial \xi}{\partial \eta} \frac{\partial \eta}{\partial m} \left[\frac{\partial \xi}{\partial \eta}\right]^T \tag{5.36}$$

$$= \frac{\partial L}{\partial m} \left[\frac{\partial \xi}{\partial \eta}\right]^T \ . \tag{5.37}$$

To insert this finding into our framework we need to transpose it:

$$FIM_{\xi^T}^{-1} \frac{\partial L}{\partial \xi^T} = \frac{\partial \xi}{\partial \eta} \frac{\partial L}{\partial m^T} = \frac{\partial \xi}{\partial \eta} \left[\frac{\partial L}{\partial m}\right]^T \ . \tag{5.38}$$

Hence, the natural gradient update rule for ξ has the following form:

$$\xi_{t+1} = \xi_t - \beta_t \frac{\partial \xi}{\partial \eta} \frac{\partial L}{\partial m^T} \ . \tag{5.39}$$

We remind ourselves that gradients of the Kullback-Leibler divergence with respect to mean parameters boils down to: $\nabla_m \mathbb{E}_{q_\eta(\theta)} \left[\log \frac{p_{\eta_0}(\theta)}{q_\eta(\theta)}\right] = \eta - \eta_0$, where η_0 represent the natural parameters of the prior. We now have a much more simplified version of the natural gradient update rule:

$$\xi_{t+1} = \xi_t - \beta_t \frac{\partial \xi}{\partial \eta} \nabla_{m^T} \mathbb{E}_{q_\xi(\theta)} \left[\log p(D \mid \theta)\right] - \beta_t \frac{\partial \xi}{\partial \eta} \left[\eta - \eta_0\right] \ . \tag{5.40}$$

5.3 Generative process of conditional sparse Student's T Processes

In this section we introduce the generative process of sparse Student-t Processes (STP), which are a variational sparse method for scaling TP to large datasets. Subsequently, we discuss some problems in performing VI over the generative process of STP and propose a solution to overcome computational difficulties by rendering the resulting ELBO inverse-free.

5.3.1 Generative process of sparse Student-t Processes

We now present our proposed generative process in the sparse case for a regression scenario:

$$\Sigma_{fu,fu} \sim \mathcal{W}_{n+m}^{-1}\left(\nu_{f,p}+n,\ \nu_{f,p}K_{fu,fu}\right) \tag{5.41}$$

$$U \mid \Sigma_{uu} \sim \mathcal{N}\left(0, \Sigma_{uu}\right) \tag{5.42}$$

$$f \mid U, \Sigma_{fu,fu} \sim \mathcal{N}\left(\Sigma_{fu}\Sigma_{uu}^{-1}U, \Sigma_{ff\cdot u}\right) \tag{5.43}$$

$$y \mid f \sim \mathcal{N}\left(f, \sigma_{noise}^2 I_n\right)\ , \tag{5.44}$$

where $\nu_{f,p} = \nu_p + m + 1$ for notation purposes so as to avoid clutter. As a consequence of Theorem 3 in Bodnar and Okhrin (2008), we can further express the following random variables which are present in equation (5.43)*:*

$$\Sigma_{ff\cdot u} \sim \mathcal{W}_{n}^{-1}\left(\nu_{f,p}+n,\ \nu_{f,p}K_{ff\cdot u}\right) \tag{5.45}$$

$$\Sigma_{uu}^{-1}\Sigma_{uf} \mid \Sigma_{ff\cdot u} \sim \mathcal{MN}\left(K_{uu}^{-1}K_{uf},\ \Sigma_{ff\cdot u} \otimes \frac{K_{uu}^{-1}}{\nu_{f,p}}\right)\ , \tag{5.46}$$

where $\otimes$ denotes the Kronecker product and $\mathcal{MN}$ denotes the matrix variate distribution. One can notice the difference in scale of the mean covariance matrix pertaining to $\Sigma_{fu,fu}$.

Differences in scaling compared to Shah et al. (2014). *To see why we have changed the scaling we need to investigate what are the first moments under the parametrization introduced in Shah et al. (2014), namely:*

$$\Sigma_{fu,fu} \sim W_{n+m}^{-1}\left(v_p+n+1, v_p K_{fu,fu}\right)\ , \tag{5.47}$$

which implies the following probabilistic formulas and their expectations:

$$\Sigma_{uu} \sim \mathcal{W}_m^{-1}\left(\nu_p + m + 1,\ \nu_p K_{uu}\right) \qquad \mathbb{E}\left[\Sigma_{uu}\right] = K_{uu} \tag{5.48}$$

$$\Sigma_{uu}^{-1} \sim \mathcal{W}_m\left(\nu_p + m + 1,\ \frac{1}{\nu_p} K_{uu}^{-1}\right) \qquad \mathbb{E}\left[\Sigma_{uu}^{-1}\right] = \frac{m + 1 + \nu_p}{\nu_p} K_{uu}^{-1} \tag{5.49}$$

$$\Sigma_{f_k f_k \cdot u} \sim Gamma^{-1}\left(\frac{\nu_p + m + 2}{2},\ \frac{\nu_p}{2} K_{f_k f_k \cdot u}\right) \quad \mathbb{E}\left[\Sigma_{f_k f_k \cdot u}\right] = \frac{\nu_p}{\nu_p + m} K_{f_k f_k \cdot u} \tag{5.50}$$

We can notice that for increasingly larger number of inducing points, the expectations over Σ_{uu}^{-1} and $\Sigma_{ff \cdot u}$ get increasingly upscaled and downscaled respectively with respect to their true solution. For $\nu_p \to \infty$, the expectations will converge towards the real solution and their variance will collapse to zero.

Under our proposed generative process:

$$\Sigma_{fu,fu} \sim \mathcal{W}_{n+m}^{-1}\left(\nu_p + m + n + 1,\ \left(\nu_p + m + 1\right) K_{fu,fu}\right) \tag{5.51}$$

we have the following distributions and expectations:

$$\Sigma_{uu} \sim \mathcal{W}_m^{-1}\left(\nu_p + m + 1,\ \left(\nu_p + m + 1\right) K_{uu}\right) \qquad \mathbb{E}\left[\Sigma_{uu}\right] = \frac{\nu_p + m + 1}{\nu_p} K_{uu} \tag{5.52}$$

$$\Sigma_{uu}^{-1} \sim \mathcal{W}_m\left(\nu_p + m + 1,\ \frac{1}{\nu_p + m + 1} K_{uu}\right) \qquad \mathbb{E}\left[\Sigma_{uu}^{-1}\right] = K_{uu}^{-1} \tag{5.53}$$

$$\Sigma_{f_k f_k \cdot u} \sim Gamma^{-1}\left(\frac{\nu_p + m + 2}{2},\ \frac{\nu_p + m + 1}{2} K_{f_k f_k \cdot u}\right) \quad \mathbb{E}\left[\Sigma_{f_k f_k \cdot u}\right] = \frac{\nu_p + m + 1}{\nu_p + m} K_{f_k f_k \cdot u} \tag{5.54}$$

In stark contrast to the parametrization introduced in Shah et al. (2014), we now have an upscaling of expectations over covariance matrices, which only reverts to the real solution for very large values of ν_p. However, in the interest of obtaining predictions at testing time from our sparse Student-T Process, we are only interested in obtaining unbiased samples from Σ_{uu}^{-1} and $\Sigma_{f_k f_k \cdot u}$, for which we can notice that our scaling factor is more appropriate.

Remark 5.3.1. *From the generative process we can see that we have matrix inversion operations in equations* (5.45) *and* (5.46)*. Subsequently, we will show how to avoid taking matrix inverses in posterior space.*

5.3.2 Prior conditional matching in covariance matrices

With the aim of obtaining a fully inverse-free method we can propose the following generative process in posterior space for the sparse Student-t Process in a regression scenario:

$$q\left(\Sigma_{fu,fu}\right) \sim \mathcal{W}_{n+m}^{-1}\left(\nu_{f,p}+n,\ \nu_{f,p}\tilde{K}_{fu,fu}\right) \tag{5.55}$$

$$q\left(U\right) \sim \mathcal{N}\left(m_U, S_U\right) \tag{5.56}$$

$$f \mid U, \Sigma_{fu,fu} \sim \mathcal{N}\left(\Sigma_{fu}\Sigma_{uu}^{-1}U, \Sigma_{ff\cdot u}\right) \tag{5.57}$$

$$y \mid f \sim \mathcal{N}\left(f, \sigma^2 I_n\right)\ , \tag{5.58}$$

where we again take $\nu_{f,p} = \nu_p + m + 1$ for notation purposes and for the time being we assume that the variational parameter $\tilde{K}_{fu,fu} = \begin{pmatrix} \tilde{K}_{uu} & K_{uf} \\ K_{fu} & \tilde{K}_{ff} \end{pmatrix}$ (see Appendix C.1 for detailed form) represents a p.s.d. matrix from which we can easily sample $\Sigma_{fu}\Sigma_{uu}^{-1}$ and $\Sigma_{ff\cdot u}$. Supposing our approximate inference method of choice is VI, then the resulting ELBO (see Appendix C.2 for a full derivation) will have the following KL-divergence between the prior and posterior over partitioned covariance matrices $\mathrm{KL}\left[q\left(\Sigma_{fu,fu}\right) \parallel p\left(\Sigma_{fu,fu}\right)\right]$, which has an analytic formula for Inverse Wishart distributions (Granström and Orguner 2012):

$$\frac{\nu_q}{2}\log\left|\tilde{K}_{fu,fu}\right| - \frac{\nu_p}{2}\log\left|K_{fu,fu}\right| + \sum_{j=1}^{n+m}\left[\log\Gamma\left(\frac{\nu_p+1-j}{2}\right) - \log\Gamma\left(\frac{\nu_q+1-j}{2}\right)\right] \tag{5.59}$$

$$+ \frac{\nu_p-\nu_q}{2}\left[\log\left|\tilde{K}_{fu,fu}\right| - \sum_{j=1}^{n+m}\psi_0\left(\frac{\nu_q+1-j}{2}\right)\right] + \mathrm{Tr}\left[-\frac{\nu_q}{2}\tilde{K}_{fu,fu}^{-1}\left(\tilde{K}_{fu,fu} - K_{fu,fu}\right)\right]\ .$$

Throughout the remainder of this chapter models using this generative process and variational parametrization of $\tilde{K}_{fu,fu}$ will be denoted as Inverse Free Student-t Processes (IF-STP).

Remark 5.3.2. *Hence the problematic part is in computing the inverse of $\tilde{K}_{fu,fu}$ (difficulty pending on parametrization form (see Appendix C.2 for discussion) and directly computing $\log|K_{fu,fu}|$, which would incur a $\mathcal{O}\left((N+M)^3\right)$ computational cost. The avoidance of these computationally expensive operations is the primary motivation for which we might not want to perform inference over this particular generative process.*

Going forward, one simple solution to eliminate these operations altogether is to impose prior conditional matching between $q\left(\Sigma_{fu,fu}\right)$ and $p\left(\Sigma_{fu,fu}\right)$, hence eliminating this KL-divergence from the resulting ELBO. The remainder of this subsection deals with this aspect.

We seek to achieve prior conditional matching in the Inverse Wishart prior over covariance matrices encompassing both training points and inducing points. First we need to remind ourselves what do we actually mean by prior conditional matching.

Brief recap on conditional prior matching. *This concept was first used in the derivation of the collapsed lower bound specific to Titsias (2009) in subsection 2.2.2, where we were tasked with augmenting the space of function values F with U. We choose the approximate posterior to be factorized as $p(F, U \mid y) = p(F \mid U, Y)p(U \mid Y)$, thus when we plugged this form into the variational lower bound on the log marginal likelihood we obtained the following cancellation in terms:*

$$\log p(y) \geq \int p(F \mid U)q(U) \log \frac{p(y \mid F)\cancel{p(F \mid U)}p(U)}{\cancel{p(F \mid U)}p(U)} \, dF \, dU \, ,$$

thus avoiding having to compute a prospective $\mathrm{KL}\left[q(F \mid U) \| p(F \mid U)\right]$, which would have arisen if we choose another form for $q(F \mid U)$. Hence, this example motivates the naming behind "prior conditional matching", as in matching the conditional posterior with the conditional prior in an augmented space.

After this brief detour, we return to our goal, which is to eliminate the computational burden stemming from having a prospective $\mathrm{KL}\left[q\left(\Sigma_{fu,fu}\right) \| \, p\left(\Sigma_{fu,fu}\right)\right]$. This can be achieved by augmenting the space of covariance functions with a random variable T and imposing prior conditional matching between $q(\Sigma_{fu,fu} \mid T)$ and $p(\Sigma_{fu,fu} \mid T)$. More precisely, this can be done by adding another layer in our Bayesian hierarchical generative process, namely:

$$T \sim \mathcal{W}_m \left(\nu_{f,p}^{W}, \, \frac{1}{\nu_{f,p}^{W}} K_{uu}^{-1} \right) \tag{5.60}$$

$$\Sigma_{fu,fu} \mid T \sim \mathcal{W}_{n+m}^{-1} \left(\nu_{f,p}^{W^{-1}} + n, \, \nu_{f,p}^{W^{-1}} \begin{bmatrix} T^{-1} & K_{uf} \\ K_{fu} & \overline{K_{ff \cdot u}} + K_{fu} T K_{uf} \end{bmatrix} \right) \tag{5.61}$$

$$U \mid \Sigma_{uu} \sim \mathcal{N}\left(0, \Sigma_{uu}\right) \tag{5.62}$$

$$f \mid U, \Sigma_{fu,fu}, T \sim \mathcal{N}\left(\Sigma_{fu}\Sigma_{uu}^{-1}U, \Sigma_{ff \cdot u}\right) \tag{5.63}$$

$$y \mid f \sim \mathcal{N}\left(f, \sigma_{noise}^2 I_n\right) \, , \tag{5.64}$$

where $\nu_{f,p}^{W} = \nu_{p}^{W} + m + 1$ and $\nu_{f,p}^{W^{-1}} = \nu_{p}^{W^{-1}} + m + 1$ for notation purposes to avoid clutter. ν_{p}^{W} and $\nu_{p}^{W^{-1}}$ are free floating hyperparameters of the Wishart and Inverse Wishart distribution respectively, both constrained to be larger than zero. We denote $\overline{K_{ff \cdot u}} = K_{ff} + K_{fu}TK_{uu}TK_{uf} - 2K_{fu}TK_{uf} \geq K_{ff \cdot u}$, with equality if and only if $T = K_{uu}^{-1}$ (Davies 2015). As a consequence of Theorem 3 in Bodnar and Okhrin (2008), we can further express the random variables present in equation (5.63) *as:*

$$\Sigma_{ff \cdot u} \mid T \sim \mathcal{W}_{n}^{-1} \left(\nu_{f,p}^{W^{-1}} + n, \, \nu_{f,p}^{W^{-1}} \overline{K_{ff \cdot u}} \right) \tag{5.65}$$

$$\Sigma_{uu}^{-1}\Sigma_{uf} \mid \Sigma_{ff \cdot u}, T \sim \mathcal{MN}\left(TK_{uf}, \, \Sigma_{ff \cdot u} \otimes \frac{T}{\nu_{f,p}} \right) \, , \tag{5.66}$$

from which we can notice that it is easy to sample from as the parameters do not involve any matrix

inversion operations.

On the choice of $\Sigma_{fu,fu} \mid T$. *From basic properties of the Wishart distribution, we know that $\mathbb{E}[T] = K_{uu}^{-1}$. With this in mind, we also have that the scaling matrix of $\Sigma_{fu,fu} \mid \mathbb{E}[T]$ results in*

$$\begin{pmatrix} (K_{uu}^{-1})^{-1} & K_{uf} \\ K_{fu} & K_{ff} + \underbrace{K_{fu}\cancel{K_{uu}^{-1}}\cancel{K_{uu}}K_{uu}^{-1}K_{uf}}_{\cancel{Q_{ff}}} - \underbrace{2K_{fu}K_{uu}^{-1}K_{uf}}_{\cancel{2Q_{ff}}} + \underbrace{K_{fu}K_{uu}^{-1}K_{uf}}_{\cancel{Q_{ff}}} \end{pmatrix} = \begin{pmatrix} K_{uu} & K_{uf} \\ K_{fu} & K_{ff} \end{pmatrix}.$$

Therefore, we have recovered the scaling matrix from the generative process of STP given in equation (5.41) *considering the expectation over T. This can be summarized in the following remark.*

Remark 5.3.3. *We notice that f no longer follows a $\mathcal{MVT}\left(\nu_p^{W^{-1}},\, 0,\, K_{ff}\right)$ distribution. Instead we have $f \mid T \sim \mathcal{MVT}\left(\nu_{f,p}^{W^{-1}},\, 0,\, \overline{K_{ff\cdot u}} + K_{fu}TK_{uf}\right)$, which in turn converges to $\mathcal{MVT}\left(\nu_{f,p}^{W^{-1}},\, 0,\, K_{ff}\right)$ as $\nu_{f,p}^{W} \to \infty$ and implicitly $T \approx K_{uu}^{-1}$.*

5.3.3 Variational Posteriors over Inverse-Wishart Processes

To perform inference over our proposed hierarchical Bayesian model, we need a way of defining a valid variational mean covariance matrix for the posterior Inverse-Wishart Process over the space of possibly infinitely large covariance matrices: $q(\Sigma_{fu,fu}) = \mathcal{W}_{n+m}^{-1}\left(\nu_{f,p}^{W^{-1}} + n, \nu_{f,p}^{W^{-1}}\tilde{K}_{fu,fu}\right)$, where $\tilde{K}_{fu,fu} \succ 0$ positive definite matrix defined as $\tilde{K}_{fu,fu} = \begin{pmatrix} T^{-1} & K_{uf} \\ K_{fu} & \overline{K_{ff\cdot u}} + K_{fu}TK_{uf} \end{pmatrix}$. To see this is true, we make use of the following proposition.

Proposition 5.3.1 (Proposition 2.1 in Gallier et al. (2010))**.** *For any symmetric matrix M of the form: $M = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$ and if C is invertible then $M \succ 0 \iff A \succ 0$ and $A - BC^{-1}B^T \succ 0$.*

Remark 5.3.4. *We note that $T^{-1} \succ 0$ and $\overline{K_{ff\cdot u}} \succ 0$ by construction, thereby $\tilde{K}_{fu,fu} \succ 0$, so that it satisfies the condition for defining mean covariance matrices for Inverse Wishart distributions. This will ensure that samples of important random variables such as $\Sigma_{fu,fu}$, $\Sigma_{ff\cdot u}$ and Σ_{uu} will be p.s.d. matrices as well.*

5.4 Evidence lower bound of conditional sparse Student-t Processes

In this section we are interested in obtaining a variational lower bound for the following posterior $q(F, U, \Sigma_{fu,fu}, T) = p(F \mid U, \Sigma_{fu,fu})\, q(U)\, p(\Sigma_{fu,fu} \mid T) q(T)$. We commence to derive the ELBO from scratch following the derivations initially introduced in Titsias (2009). We consider

our training data $D = \{X_i, Y_i\}_{i=1,\cdots,n}$. *Our goal is to approximate the true posterior distribution* $p(F, U, \Sigma_{fu,fu}, T \mid D)$, *which translates to minimizing the following:*

$$= \int Q(F, U, \Sigma_{fu,fu}, T) \log \frac{Q(F, U, \Sigma_{fu,fu}, T)}{p(F, U, \Sigma_{fu,fu}, T|D)} \mathrm{d}F \, \mathrm{d}U \, \mathrm{d}\Sigma_{fu,fu} \, \mathrm{d}T \tag{5.67}$$

$$= \int Q(F, U, \Sigma_{fu,fu}, T) \left[\log \frac{Q(F, U, \Sigma_{fu,fu}, T)}{p(F, U, \Sigma_{fu,fu}, T, D)} + \log P(D) \right] \mathrm{d}F \, \mathrm{d}U \, \mathrm{d}\Sigma_{fu,fu} \, \mathrm{d}T \tag{5.68}$$

$$= \mathbb{E}_{Q(F,U,\Sigma_{fu,fu},T)} \left[\log \frac{Q(F, U, \Sigma_{fu,fu}, T)}{p(D|F, U, \Sigma_{fu,fu}, T) p(F, U, \Sigma_{fu,fu}, T)} + \log P(D) \right] , \tag{5.69}$$

which after some rearrangements becomes:

$$\begin{aligned} \log p(D) - [&\mathbb{E}_{q(F,U,\Sigma_{fu,fu},T)} \log p(D|F, U, \Sigma_{fu,fu}, T) \\ &- \mathrm{KL}\left[Q(F, U, \Sigma_{fu,fu}, T) \| p(F, U, \Sigma_{fu,fu}, T)\right]] \end{aligned} \tag{5.70}$$

We can now re-express this derivation as:

$$\begin{aligned} \log p(D) - \mathrm{KL}\left[q(F, U, \Sigma_{fu,fu}, T) \| p(F, U, \Sigma_{fu,fu}, T)\right] = \\ \mathbb{E}_{q(F,U,\Sigma_{fu,fu},T)} \log p(D|F, U, \Sigma_{fu,fu}, T) \\ - \mathrm{KL}\left[Q(F, U, \Sigma_{fu,fu}, T) \| p(F, U, \Sigma_{fu,fu}, T)\right] \end{aligned} \tag{5.71}$$

Hence, the standard variational approach (Jordan et al. 1999) translates minimizing the KL divergence $KL[q(F, U, \Sigma_{fu,fu}, T) \| p(F, U, \Sigma_{fu,fu}, T|D)]$, *which is intractable, to essentially maximizing a lower bound on the log marginal likelihood that is tractable, respectively the r.h.s. of the previous equation:*

$$\begin{aligned} \mathcal{L}_{IF-CSTP} = \mathbb{E}_{Q(F,U,\Sigma_{fu,fu},T)} \log p(Y \mid F, U, \Sigma_{fu,fu}, T) - \\ \int q(F, U, \Sigma_{fu,fu}, T) \log \frac{q(F, U, \Sigma_{fu,fu}, T)}{p(F, U, \Sigma_{fu,fu}, T)} \mathrm{d}F \, \mathrm{d}U \, \mathrm{d}\Sigma_{fu,fu} \, \mathrm{d}T \, , \end{aligned} \tag{5.72}$$

where we now define the constituent terms of the factorized approximate posterior as follows:

$$q(T) = \mathcal{W}_m \left(T \mid \nu^W_{f,q}, \ \frac{1}{\nu^W_{f,q}} L_{\tilde{K}^{-1}_{uu}} L^\top_{\tilde{K}^{-1}_{uu}} \right) \tag{5.73}$$

$$q(\Sigma_{fu,fu} \mid T) = \mathcal{W}^{-1}_{n+m} \left(\Sigma_{fu,fu} \mid \nu^{W^{-1}}_{f,q} + n, \ \nu^{W^{-1}}_{f,q} \begin{bmatrix} T^{-1} & K_{uf} \\ K_{fu} & \overline{K_{ff \cdot u}} + K_{fu} T K_{uf} \end{bmatrix} \right) \tag{5.74}$$

$$q(U) = \mathcal{N}\left(U \mid m_U, L_{S_U} L^\top_{S_U}\right) \tag{5.75}$$

$$q(f \mid U, \Sigma_{fu,fu}) = \mathcal{N}\left(f \mid \Sigma_{fu} \Sigma^{-1}_{uu} U, \Sigma_{ff \cdot u}\right) \ . \tag{5.76}$$

Throughout the remainder of this chapter this model will be denoted as Inverse Free Conditional Student-t Processes (IF-CSTP). We now proceed to examine each term of the ELBO specific to this model introduced in equation (5.72)*.*

Expectation over data fit term in SVI setting. *We can re-express expectations over individual terms as follows:*

$$\sum_{i=1}^{n} E_{q(\Sigma_{f_i u, f_i u})} \left[\int \log p(Y_i \mid F_i) q(F_i; \tilde{U}(\Sigma_{f_i u, f_i u}), \tilde{\Sigma}(\Sigma_{f_i u, f_i u}))\, dF_i \right] ,$$

where $\tilde{U}(\Sigma_{f_i u, f_i u}) = \Sigma_{f_i u} \Sigma_{uu}^{-1} m_U$ *and* $\tilde{\Sigma}(\Sigma_{f_i u, f_i u}) = \Sigma_{f_i f_i} - \Sigma_{f_i u} \Sigma_{uu}^{-1} \left[\Sigma_{uu} - S_U \right] \Sigma_{uu}^{-1} \Sigma_{u f_i}$ *define distributions over scalars. Hence, the multivariate formulas introduced in equations* (5.65) *and* (5.66) *can now be rewritten as:*

$$\Sigma_{f_i f_i \cdot u} \mid T \sim Gamma^{-1} \left(\frac{\nu_{f,q}^{W^{-1}} + 1}{2}, \frac{\nu_{f,q}^{W^{-1}}}{2} \overline{K_{f_i f_i \cdot u}} \right) \tag{5.77}$$

$$\Sigma_{uu}^{-1} \Sigma_{u f_i} \mid \Sigma_{f_i f_i \cdot u}, T \sim \mathcal{N} \left(T K_{u f_i}, \frac{\Sigma_{f_i f_i \cdot u}}{\nu_{f,q}^{W^{-1}}} \otimes T \right) . \tag{5.78}$$

We can notice that we can easily sample from both distributions without incurring any computational burden.

Kullback-Leibler Divergences. *In the KL term present in our lower bound we can separate the fraction terms with respect to integrants that they exclusively depend on:*

$$\begin{aligned} \mathbb{E}_{q(F,U,\Sigma_{fu,fu},T)} \log \frac{\cancel{q(F|U,\Sigma_{fu,fu},T)}}{\cancel{p(F|U,\Sigma_{fu,fu},T)}} + \mathbb{E}_{q(\Sigma_{uu}^{-1},T)} \int q(U) \log \frac{q(U)}{p(U|\Sigma_{uu}^{-1})} \mathrm{d}U \\ + \mathbb{E}_{q(T)} \mathbb{E}_{q(\Sigma_{fu,fu}|T)} \log \frac{\cancel{q(\Sigma_{fu,fu} \mid T)}}{\cancel{p(\Sigma_{fu,fu} \mid T)}} + \mathbb{E}_{q(T)} \log \frac{q(T)}{p(T)} , \end{aligned} \tag{5.79}$$

where the first and third terms cancel out due to conditional prior matching in both function and covariance space, the latter cancellation being our objective defined in subsection 5.3.2. We focus now on the remaining two terms.

KL-divergence between inducing points. *The term* $\mathbb{E}_{q(\Sigma_{uu}^{-1},T)} \int q(U) \log \frac{q(U)}{p(U|\Sigma_{uu}^{-1})} \mathrm{d}U$ *expands as:*

$$\mathbb{E}_{q(T)} \mathbb{E}_{q(\Sigma_{uu}^{-1}|T)} \frac{1}{2} \left[-\log \left| \Sigma_{uu}^{-1} \right| - \log |S_U| - d + \mathrm{Tr} \left[\Sigma_{uu}^{-1} S_U \right] + m_U^\top \Sigma_{uu}^{-1} m_U \right] . \tag{5.80}$$

We remind ourselves that we can obtain samples from Σ_{uu}^{-1} via following Bayesian hierarchical construction:

$$q\left(T\right)=\mathcal{W}_m\left(\nu_{f,q}^W,\ \frac{1}{\nu_{f,q}^W}L_{\tilde{K}_{uu}^{-1}}L_{\tilde{K}_{uu}^{-1}}^\top\right);\quad q\left(\Sigma_{uu}^{-1}\mid T\right)=\mathcal{W}_m^{-1}\left(\nu_{f,q}^{W^{-1}},\ \frac{1}{\nu_{f,q}^{W^{-1}}}T\right)\ . \tag{5.81}$$

No matrix inversion is required for this divergence term. Moreover, T is obtained via the Bartlett decomposition $L_{\tilde{K}_{uu}^{-1}}A_TA_T^\top L_{\tilde{K}_{uu}^{-1}}^\top$, where A_T is a lower triangular matrix defined in subsection 5.2.1. Then, we can apply again the Bartlett decomposition to obtain samples from Σ_{uu}^{-1} by noticing that $L_{\tilde{K}_{uu}^{-1}}A_T$ represents a lower triangular matrix. Finally, samples can be obtained with the following construction: $L_{\tilde{K}_{uu}^{-1}}A_TA_{\Sigma_{uu}^{-1}}A_{\Sigma_{uu}^{-1}}^\top A_T^\top L_{\tilde{K}_{uu}^{-1}}^\top$ Hence, Σ_{uu}^{-1} is a product of lower triangular matrices $L_{\tilde{K}_{uu}^{-1}}A_TA_{\Sigma_{uu}^{-1}}$, with the determinant $\log|\Sigma_{uu}^{-1}|$ having an analytic formula.

KL-divergence between inducing points' inverse covariance matrices. *We now focus on the term $\mathrm{KL}\left[q(T)\|p(T)\right]$, which has an analytic formula:*

$$\begin{aligned}-\frac{\nu_{f,p}^W}{2}\left[\log\left|\nu_{f,p}^WK_{uu}\right|+\log\left|\frac{1}{\nu_{f,q}^W}\tilde{K}_{uu}^{-1}\right|\right]+\frac{\nu_{f,q}^W}{2}\left[\mathrm{Tr}\left[\frac{\nu_{f,p}^W}{\nu_{f,q}^W}K_{uu}\tilde{K}_{uu}^{-1}\right]-M\right]\\+\log\frac{\Gamma_m(\nu_{f,p}^W/2)}{\Gamma_m(\nu_{f,q}^W/2)}+\frac{\nu_{f,q}^W-\nu_{f,p}^W}{2}\psi_m(\frac{\nu_{f,q}^W}{2})\ ,\end{aligned} \tag{5.82}$$

where ψ_m denotes the multivariate digamma function (the derivative of the log of the multivariate gamma function). From the above equation we can notice that no matrix inverses are required in the computation of the KL divergence. Directly computing $\log|K_{uu}|$ would incur an $\mathcal{O}\left(M^3\right)$ computational cost. To bypass it, we use the following proposition.

Proposition 5.4.1. *For a p.s.d. matrix $K\in M_{n\times n}$, x being a conjugate gradient solution to $Kx=g$, where $g\in M_{n\times m}$ are independent standard normal samples and $\tilde{K}\succ 0$ we have the following lower bound on the log determinant of K:*

$$log|K|\geq n+\frac{1}{m}\sum_{i=1}^{m}-g_i^\top\tilde{K}x_i+log\left|\tilde{K}\right| \tag{5.83}$$

with the bound being tight if and only if $\tilde{K}\approx K$.

Proof. We start with a lower bound on $\log|K|$, which can be expressed as:

$$log|K|\geq\mathrm{Tr}\left[I_n-K^{-1}\right]\ . \tag{5.84}$$

Hutchinson (1989) introduced a stochastic estimation method for computing matrix traces, which

is defined as follows:

$$H_m(A) = \frac{1}{m}\sum_{i=1}^{m} g_i^\top A g_i \ , \tag{5.85}$$

which was shown to converge towards $\mathrm{Tr}(A)$ as $m \to \infty$, where $A \in M_{n\times n}$. Earlier work by Girard (1987) suggests taking $g_i \in M_{n\times 1}$ with individual elements of the column matrix stemming from standard normal, whereas Hutchinson (1989) advocated for taking the individual elements as Rademacher random variables, which translates into equally probable $\{-1, 1\}$ splits.

In light of this, we can rewrite our objective as follows $\log|K| = \log\left|K\tilde{K}^{-1}\right| + \log\left|\tilde{K}\right|$ for any arbitrary p.s.d. matrix $\tilde{K}$. We are interested in obtaining reliable estimates of the following bound without performing the matrix inversion operation:

$$log\left|K\tilde{K}^{-1}\right| + log\left|\tilde{K}\right| \geq \mathrm{Tr}\left[I_n\right] - \mathrm{Tr}\left[\left(K\tilde{K}^{-1}\right)^{-1}\right] + log\left|\tilde{K}\right| \tag{5.86}$$

$$\geq \mathrm{Tr}\left[I_n\right] - \frac{1}{m}\sum_{i=1}^{m} g_i^\top \left(K\tilde{K}^{-1}\right)^{-1} g_i + log\left|\tilde{K_{fu,fu}}\right| \ . \tag{5.87}$$

Instead of obtaining lower/upper bounds on the part involving matrix inversion in the above equation, we can reliably estimate the solution via conjugate gradients. The solution will converge in one iteration if the condition number is close to 1, which will happen if $\tilde{K}^{-1}$ is an accurate estimate of K^{-1}.

We briefly summarize the theory behind conjugate gradients and how to compute them. We are interested in solution to $Kx = y$, where the optimal solution $x^{opt} = K^{-1}y$, where $K \in M_{n\times n}$ and $y \in M_{n\times 1}$. Conjugate gradients start with an initial solution x_0, which progressively gets refined and is guaranteed to converge in the worst case scenario in n iterations. We provide the pseudocode for Preconditioned Conjugate Gradients (PCG) in algorithm 3. For an in-depth introduction to conjugate gradients we refer the reader to Davies (2015).

For the purposes of our problem at hand, we have $y = g_i$ and the CG-based solution $x_i \approx K^{-1}g_i$. Then, the lower bound from equation (5.87) translates to:

$$log|K| \geq \mathrm{Tr}\left[I_n\right] - \frac{1}{m}\sum_{i=1}^{m} g_i^\top \tilde{K} x_i + log\left|\tilde{K}\right| \ , \tag{5.88}$$

which proves our proposition. Furthermore, the tightness of the lower bound in the case $\tilde{K} \approx K$

Algorithm 3: Preconditioned Conjugate Gradient routine

Input: $K \in M_{n\times n}, y \in M_{n\times 1}$
Preconditioning matrix: $M \in M_{n\times n}$
Initial estimate: $x_0 \in M_{n\times 1}$
Compute first search direction and basis
$r_0 = Kx_0 - y$
$z_0 = M^{-1}r_0$
$p_0 = z_0$
for $i = 1$ **to** n **do**
  Alpha: $\alpha_i = \frac{r_i^\top r_i}{p_i^\top K p_i}$
  Solution update: $x_{i+1} = x_i + \alpha_i p_i$
  if i=n **then**
    return x_i
  end if
  $z_{i+1} = M^{-1}r_{i-1}$
  Beta: $\beta_i = \frac{r_{i+1}^\top r_{i+1}}{r_i^\top r_i}$
  Search direction update: $p_{i+1} = z_{i+1} - \beta_i p_i$
end for

can be shown as follows:

$$log|K| \geq \mathrm{Tr}\left[I_n\right] - \frac{1}{m}\sum_{i=1}^{m} g_i^\top \tilde{K} x_i + log\left|\tilde{K}\right| \tag{5.89}$$

$$log|K| \geq \mathrm{Tr}\left[I_n\right] - \frac{1}{m}\sum_{i=1}^{m} g_i^\top \tilde{K} K^{-1} g_i + log\left|\tilde{K}\right| \tag{5.90}$$

$$log|K| \geq \mathrm{Tr}\left[I_n\right] - \mathrm{Tr}\left[I_n\right] + \log\left|\tilde{K}\right| \tag{5.91}$$

$$log|K| = \log\left|\tilde{K}\right|, \tag{5.92}$$

where we used the fact that $x_i = K^{-1}g_i$ since the condition number is 1 in this scenario. □

Remark 5.4.1. *Translating Proposition 5.4.1 to our case, we have* $log|K_{uu}| \geq n + \frac{1}{m}\sum_{i=1}^{m} -g_i^\top T^{-1} x_i - log|T|$. *In the computation of* x_i *we precondition with* T. *Hence if* $T \approx K_{uu}^{-1}$ *the preconditioned CG routine will finish in one step. We empirically prove on UCI datasets that this method is close to* $\mathcal{O}\left(M^2\right)$ *(see Figure 5.9) due to preconditioning. The worst case scenario complexity is* $\mathcal{O}\left(M^3\right)$, *which is similar to the computational complexity of performing Cholesky decomposition in the standard SVGP ELBO. However, this never occurred in our experiments. Moreover, whereas in current work we use standard preconditioned conjugate gradients, one could use the Unbiased LInear Systems SolvEr (ULISSE) (Filippone and Engler 2015a), a randomly truncated CG run, to compute unbiased estimate of* x_i.

5.5 Natural Gradients eliminate iterative bounds

Motivation. *The proposed ELBO in subsection 5.4 involves an iterative PCG lower bound on* $\log|K_{uu}|$. *In the case that* $\tilde{K}_{uu}^{-1}$ *diverges from its exact solution, then the number of PCG iterations will increase to the point of passing our computational budget. Moreover, even for low tolerance values set for the residual vector, this will not guarantee that our resulting ELBO will still constitute an actual lower bound on the log marginal likelihood. Therefore, instead of using iterative-based lower bounds on log determinants present in the KL-divergence, we can bypass these approximations by taking natural gradients of variational parameters.*

We remind ourselves of the general natural gradient descent (NGD) update rule introduced in subsection 5.2.3:

$$\xi_{t+1} = \xi_t - \beta_t \underbrace{\frac{\partial \xi}{\partial \eta} \nabla_{m^T} \mathbb{E}_{q_\xi(\theta)} \left[\log p(D \mid \theta)\right]}_{\textit{Data Fit NGD}} - \beta_t \underbrace{\frac{\partial \xi}{\partial \eta} \left[\eta - \eta_0\right]}_{\textit{KL-div. NGD}} . \tag{5.93}$$

In section 5.4 we have seen that no computationally expensive operations were necessary for the KL divergence pertaining to U, *but the computation of* $\log|K_{uu}|$ *stemming from the KL divergence associated to* T *requires an iterative PCG routine. In this section we shall see that the natural gradients of* U *and* T *do not involve any computationally expensive operations. We commence by deriving the NGD update rules of KL divergence terms in* $\mathcal{L}_{IF-CSTP}$ *as defined in equation* (5.72)*, subsequently introducing the framework by which we compute the NGD update rules of the data fit term associated to* $\mathcal{L}_{IF-CSTP}$.

For subsequent derivations, we make use of the following useful identity from matrix differential calculus:

$$\frac{\partial X^{-1}}{\partial X_{[i,j]}} = -X^{-1} E_{[i,j]} X^{-1} , \tag{5.94}$$

where $E_{[i,j]}$ *is a matrix of zeros, except the cell at position* $[i, j]$ *being set to 1. On a similar note, the following theorem will be useful for computing explicit derivatives throughout the remainder of this section.*

Theorem 5.5.1. *(Variation of Theorem A.1 in Särkkä (2013)) For* $A = LL^\top$, *where* L *is a lower triangular matrix, we can explicitly calculate the derivative of the Cholesky factorization w.r.t. to specific cells in matrix* A *as follows:*

$$\frac{\partial L}{\partial A_{ij}} = L\Phi\left[L^{-1} E_{ij} L^{-\top}\right] , \tag{5.95}$$

where Φ *is an operator taking the lower triangular matrix (with diagonal elements set to half their value), respectively* E_{ij} *is a matrix of zeros, except at positions* (i, j) *and* (j, i), *which are set to 1.*

Natural Gradients of KL-Divergence pertaining to U. *We focus on computing derivatives of $\mathbb{E}_{q(\Sigma_{uu}^{-1},T)} \int q(U) \log \frac{q(U)}{p(U|\Sigma_{uu}^{-1})} \mathrm{d}U$, where in this case $\xi = \xi(\eta)$ and $\xi[0] = m_U$, respectively $\xi[1] = L_{S_U}$ with $L_{S_U} L_{S_U}^T = S_U$.*

Computing derivatives of explicit KL divergence w.r.t. mean parameters of U is less involved:

$$\nabla_m \mathbb{E}_{q(T)} \mathbb{E}_{q(\Sigma_{uu}^{-1}|T)} \int q(U) \log \frac{q(U)}{p(U|\Sigma_{uu}^{-1})} \mathrm{d}U = \mathbb{E}_{q(T)} \mathbb{E}_{q(\Sigma_{uu}^{-1}|T)} \nabla_m \mathrm{KL}\left[q(U) \| p(U|\Sigma_{uu}^{-1})\right] \tag{5.96}$$

$$= \mathbb{E}_{q(T)} \mathbb{E}_{q(\Sigma_{uu}^{-1}|T)} \left[\eta - \left(\eta_0 \mid \Sigma_{uu}^{-1}\right)\right] \tag{5.97}$$

$$= \begin{pmatrix} S_U^{-1} m_U \\ -\frac{1}{2} S_U^{-1} \end{pmatrix} - \mathbb{E}_{q(T)} \mathbb{E}_{q(\Sigma_{uu}^{-1}|T)} \left[\begin{pmatrix} 0 \\ -\frac{1}{2} \Sigma_{uu,t}^{-1} \end{pmatrix} \right] \tag{5.98}$$

$$= \begin{pmatrix} S_U^{-1} m_U \\ -\frac{1}{2} S_U^{-1} \end{pmatrix} - \begin{pmatrix} 0 \\ -\frac{1}{2} \tilde{K}_{uu}^{-1} \end{pmatrix} . \tag{5.99}$$

For a multivariate normal we choose our parametrization as $\xi = \begin{pmatrix} m_U \\ L_{S_U} \end{pmatrix}$ and we know that $\eta = \begin{pmatrix} S_U^{-1} m_U \\ -\frac{1}{2} S_U^{-1} \end{pmatrix}$. Hence, we can now derive $\frac{\partial \xi}{\partial \eta}$ as follows:

$$\frac{\partial m_U}{\partial \eta_1} = \frac{\partial S_U \eta_1}{\partial \eta_1} = S_U \tag{5.100}$$

$$\frac{\partial L_{S_U}}{\partial S_U} \frac{\partial S_U}{\partial \left[\eta_2\right]_{i,j}} = \frac{\partial L_{S_U}}{\partial S_U} \frac{\partial (-2\eta_2^{-1})}{\left[\eta_2\right]_{i,j}} \tag{5.101}$$

$$= \left[L_{S_U} \Phi \left[L_{S_U}^{-1} E_{ij} L_{S_U}^{-\top}\right]\right] \left[2\eta_2^{-1} E_{i,j} \eta_2^{-1}\right] \tag{5.102}$$

$$= \left[L_{S_U} \Phi \left[L_{S_U}^{-1} E_{ij} L_{S_U}^{-\top}\right]\right] \left[2 S_U E_{i,j} S_U\right] . \tag{5.103}$$

Natural Gradients of KL-Divergence pertaining to T. *We commence to derive the derivatives w.r.t. mean parameters m of T based on the KL divergence:*

$$\nabla_m \int q(T) \log \frac{q(T)}{p(K_{uu}^{-1})} \mathrm{d}T = [\eta - \eta_0] \tag{5.104}$$

$$= \begin{pmatrix} -\frac{\nu_{f,q}^W}{2} \tilde{K}_{uu} + \frac{\nu_{f,p}^W}{2} K_{uu} \\ \frac{\nu_{f,q}^W - m - 1}{2} - \frac{\nu_{f,p}^W - m - 1}{2} \end{pmatrix} . \tag{5.105}$$

One major advantage of the Natural Gradients framework applied to our scenario at hand is the relative ease of computing derivatives w.r.t. the KL-divergence for T. We remind ourselves that the natural parameters under our posterior formulation are:

$$\eta = \begin{pmatrix} -\frac{1}{2} \nu_{f,q}^W \tilde{K}_{uu} \\ \frac{\nu_{f,q}^W - m - 1}{2} \end{pmatrix} . \tag{5.106}$$

To compute $\frac{\partial \xi_1}{\partial \eta_1}$ we can use the chain rule:

$$\frac{\partial L_{\tilde{K}_{uu}^{-1}}}{\partial [\eta_1]_{i,j}} = \frac{\partial L_{\tilde{K}_{uu}^{-1}}}{\partial \tilde{K}_{uu}^{-1}} \frac{\partial \tilde{K}_{uu}^{-1}}{\partial \tilde{K}_{uu}} \frac{\partial \tilde{K}_{uu}}{\partial [\eta_1]_{i,j}} \tag{5.107}$$

$$= L_{\tilde{K}_{uu}^{-1}} \Phi \left[L_{\tilde{K}_{uu}}^{-1} E_{ij} L_{\tilde{K}_{uu}}^{-\top} \right] \left[-\tilde{K}_{uu}^{-1} E_{[i,j]} \tilde{K}_{uu}^{-1} \right] \left[-\frac{2}{\nu_{f,q}^W} \right] . \tag{5.108}$$

We emphasize that no costly matrix inversion operation is required to compute the above derivative. Finally, using elementary differentiation techniques we obtain that $\frac{\partial \nu_{f,q}^W}{\partial \eta_2} = 2$.

Natural Gradients of Expected Data Fit Term. *We now focus our attention on the derivation of natural gradients with respect to the expected data fit term of $\mathcal{L}_{IF-CSTP}$ (equation (5.72)) of the following distributions: $T \sim \mathcal{W}_m \left(\nu_{f,q}^W, \frac{1}{\nu_{f,q}^W} L_{\tilde{K}_{uu}^{-1}} L_{\tilde{K}_{uu}^{-1}}^\top \right)$ and $U \sim \mathcal{N} \left(m_U, L_{S_U} L_{S_U}^\top \right)$. We follow the derivation in Appendix B.1 of Khan and Lin (2017), hence for each aforementioned distribution we need to compute the following:*

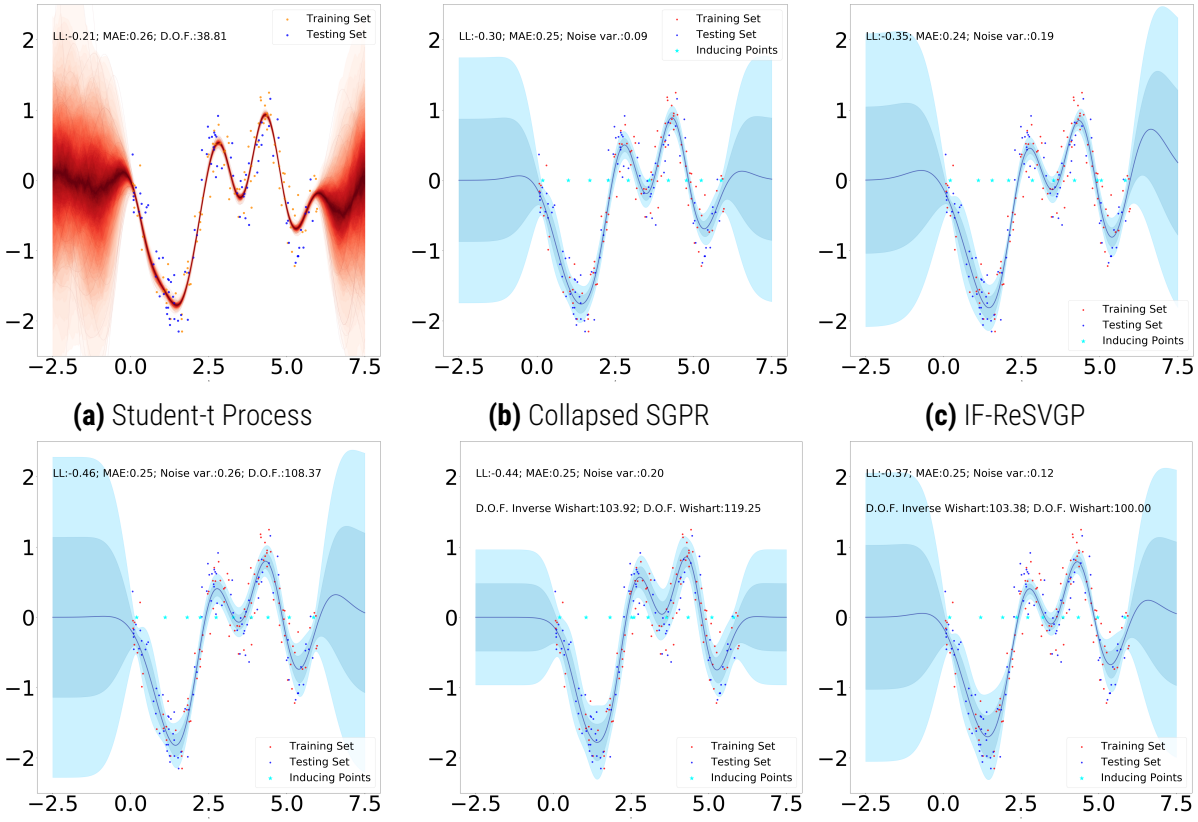
$$\frac{\partial L}{\partial m_1} = \frac{\partial L}{\partial \xi_1} \frac{\partial \xi_1}{\partial m_1} + \frac{\partial L}{\partial \xi_2} \frac{\partial \xi_2}{\partial m_1} \tag{5.109}$$

$$\frac{\partial L}{\partial m_2} = \frac{\partial L}{\partial \xi_1} \frac{\partial \xi_1}{\partial m_2} + \frac{\partial L}{\partial \xi_2} \frac{\partial \xi_2}{\partial m_2} , \tag{5.110}$$

where L represents the data fit term of $\mathcal{L}_{IF-CSTP}$. The individual results for the aforementioned equations for each distribution (i.e., $q(T)$ and $q(U)$) are provided in Appendix D.

5.6 Experiments

In this section we will devise several experiments to assess the convergence of IF-STP (i.e., model briefly introduced in subsection 5.3.2, see Appendix C for details on ELBO and approximate posterior) and IF-CSTP (natural gradient variant will be denoted as IF-CSTP-NGD) to the intended behaviour of matrix inversion dependent models such as TP (Shah et al. 2014) and collapsed SGPR (Titsias 2009) for regression and SVGP (Hensman et al. 2013) for classification. Moreover, we also compare with matrix inversion-free models such as IF-ReSVGP (Wilk et al. 2020b). We commence by performing several sensitivity analysis scenarios on toy datasets for both IF-STP and IF-CSTP, subsequently using just IF-CSTP on medium-scale and large-scale datasets from the UCI machine learning dataset repository.



(a) Student-t Process **(b)** Collapsed SGPR **(c)** IF-ReSVGP

(d) IF-STP **(e)** IF-CSTP **(f)** IF-CSTP-NGD

Figure 5.1: Convergence of inverse-free models on toy regression task. Predictive mean and variance for models trained on "snelson" dataset. Empirical mean and variance are provided for Student-t Process, based on 50 samples in function space. Likelihood variance is not added.

Convergence to Matrix Inversion based counterparts. *We are interested to find out whether our proposed models are capable to recover the testing time predictive behaviour of TP and collapsed SGPR on toy regressions tasks. For this, we use the "snelson" dataset (Snelson and Ghahramani 2006) with all sparse models using 10 inducing points, with hyperparameters initialized with the*

same values for all models considered.

IF-STP follows the predictive behaviour of collapsed SGPR more closely than IF-CSTP (Figure 5.1). Two additional remarks can be made about IF-CSTP. Firstly, the kernel variance is underestimated in comparison to both collapsed SGPR and TP. Secondly, the variance in the neighbourhood of inducing points is slightly inflated in comparison to IF-STP. This discrepancy is caused by the additional uncertainty injected by sampling from T, which results in the latent function f conditioned on T to be a multivariate Student-t distribution, albeit with higher variance in the vicinity of inducing points' locations Z. Similar comments about the increased variance in the vicinity of Z can be made for IF-CSTP-NGD, with the difference that the the optimization of the kernel variance has recovered the collapsed SGPR solution. Lastly, IF-ReSVGP has comparatively reduced uncertainty since it treats T as point-mass estimate.

Increasing the number of inducing points. *We are interested to find out how our proposed models behave given varying numbers of inducing points. For this we train all models considered on the "banana" dataset with a number of inducing points starting from 8 to 64 (Figure 5.2). A visible discrepancy emerges between SVGP and all matrix inversion-free models considered. SVGP optimized the inducing points' locations Z towards the decision boundaries, thereby enabling sharper decision boundaries. Conversely, for IF-STP the inducing points' locations Z remain stable regardless of the number of inducing points used. The reason for this relates to the large penalty inflicted by $\mathrm{KL}\left[q(\Sigma_{fu,fu})\|p(\Sigma_{fu,fu})\right]$ upon any modifications being done to either Z or kernel hyperparameters. Similar remarks can be made to IF-CSTP, albeit to a lesser degree since for this model the penalty stems from $\mathrm{KL}\left[q(T)\|p(T)\right]$. For 32 and 64 inducing points, IF-CSTP barely changes Z, whereas in the case of 8 and 16 inducing points Z are not necessarily brought towards the decision boundaries. Unsurprisingly, for IF-ReSVGP Z have a more dynamical behaviour during optimization compared to our proposed inverse-free models. This is due to the fact that it does not impose any regularization on T, instead treating it as a point-mass estimate of the real solution to K_{uu}^{-1}. Lastly, IF-CSTP-NGD induces an even more stable optimization trajectory for Z compared to IF-CSTP. This might be caused by the fact that NGD update rule for U can more easily find optima, as given by the sharper decision boundaries, so there is a lesser need to move Z along the two dimensional input space to devise decision boundaries.*

How well does T approximate K_{uu}^{-1}? *In Figure 5.3 we visualise the quality of the approximate posterior over T for the IF-CSTP models trained as part of the experiment presented in Figure 5.2. Since T has an explicit link to the matrix inverse K_{uu}^{-1}, we should expect to obtain an almost identity matrix in the case of multiplying it with K_{uu}. We can notice that this is indeed the case up to a certain degree, with occasional divergences from the expected behaviour.*

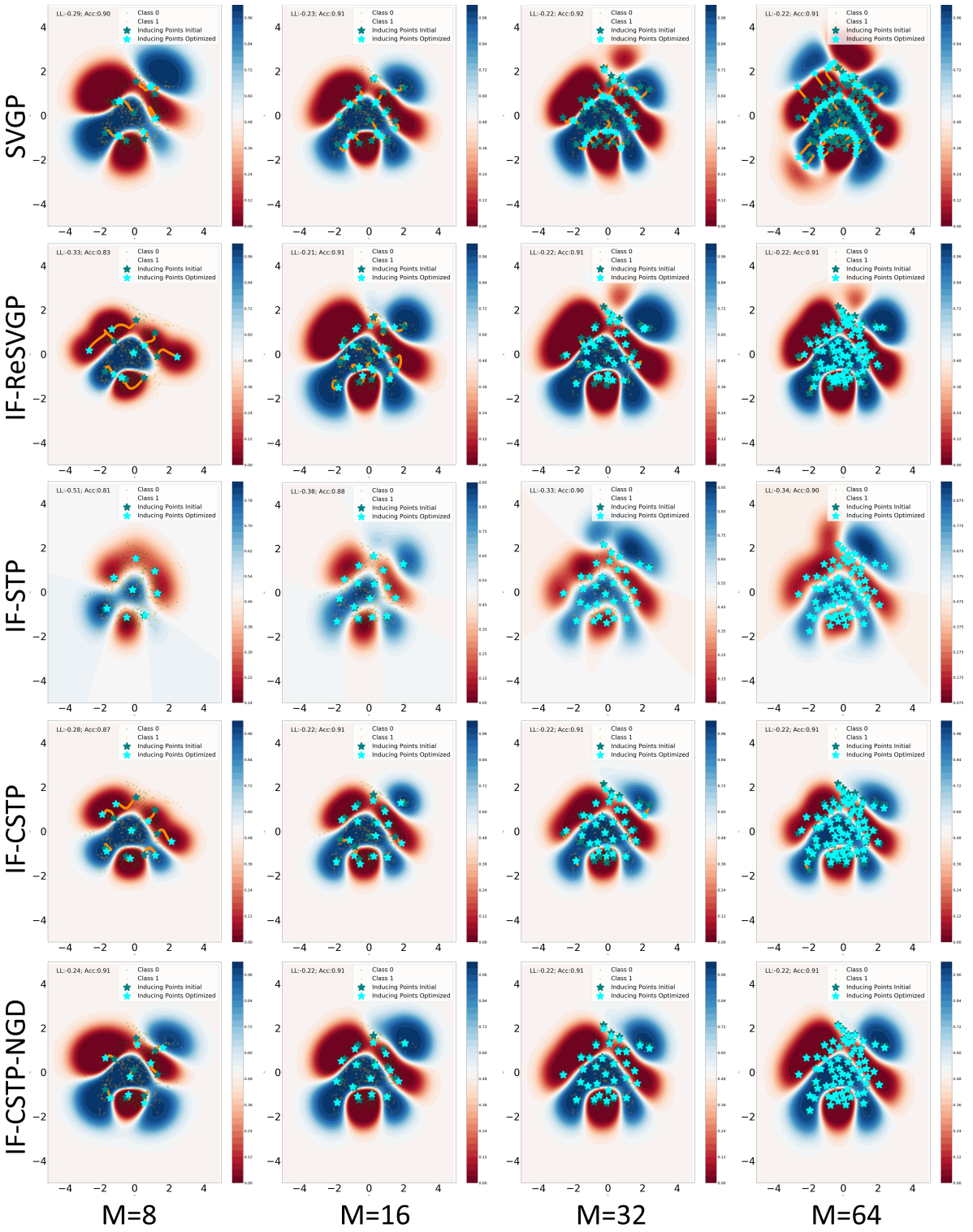


Figure 5.2: Optimization trajectory of inducing locations on toy classification task. Effects on predictive mean of increasing number of inducing points for the "banana" dataset. With dark orange we plot the optimization trajectory of Z during training from the initial teal stars to the optimized cyan stars.

Correct optimization of inducing points' locations. *The influence of the initial values of* $\nu_{f,q}^{W^{-1}}$ *(for IF-STP) and also of* $\nu_{f,q}^{W}$ *(for IF-CSTP) are of paramount importance as it linearly affects the*

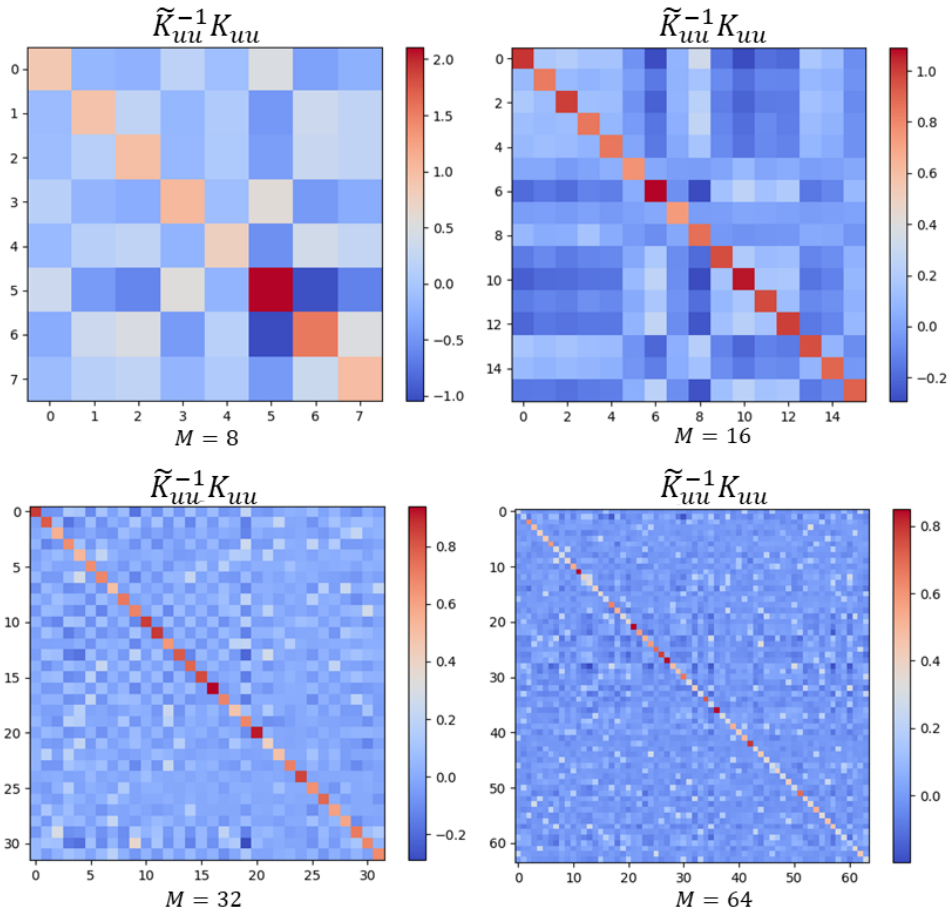
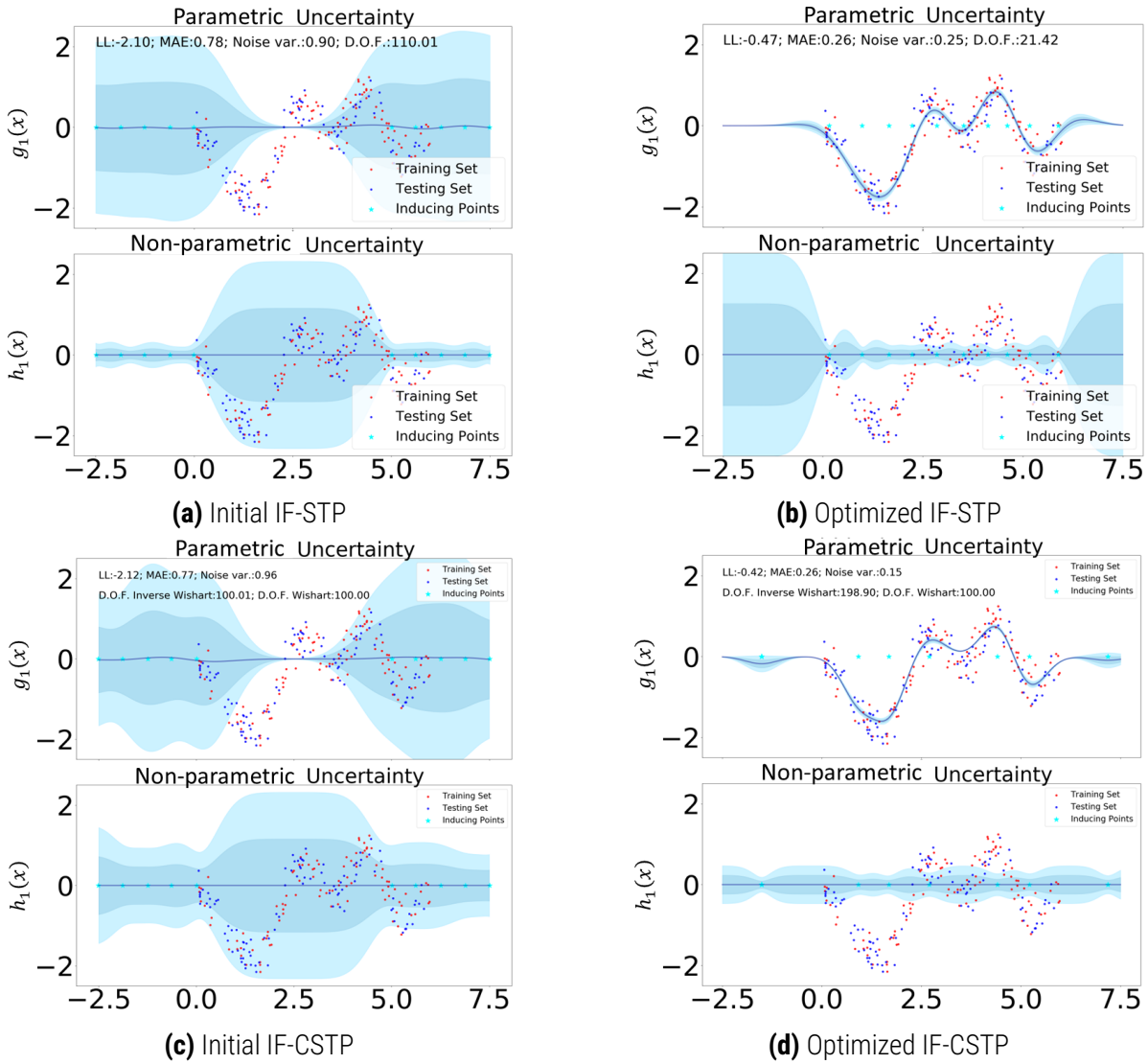


Figure 5.3: Quality of approximate posterior of T. Visualisation of $\tilde{K}_{uu}^{-1}K_{uu}$ at end of the "banana" experiment for different numbers of inducing points. The recovery of the inverse through optimisation should equate in an exact identity matrix.

size of the penalization stemming from the KL divergence, possibly leading to scenarios where the optimization prefers to keep Z and kernel hyperparameters fixed (such as in the "banana" dataset as shown in Figure 5.2). We explore this scenario and see if the inducing points' locations Z will get optimized to ideal locations despite starting from values outside the training set range. From Figure 5.4 we can easily notice that the optimized Z values are brought back to locations similar to ones obtained from running k-means in the case of IF-STP. Conversely, for IF-CSTP there are two remaining inducing points which have not been optimized to be inside the data manifold.

Lower bound recovers true approximate posteriors despite erroneous initialization. *We aim to quantify the influence of "accurate" initializations of variational parameters on the converged solution. In the case of IF-CSTP one would expect to initialize $L_{\tilde{K}_{uu}^{-1}}$ with their respective values given initial estimates of Z and kernel hyperparameters θ. Similarly for IF-STP, one can expect the same reasonable initializations for $L_{\Sigma_{uu}}$ and $L_{\Sigma_{ff\cdot u}}$ (see Appendix C for details). We instead initialize them with identity matrices to determine if the bound can recover optimal posteriors. The motivation behind this experiment is to test if variational parameters can be reliably optimized through our ELBOs*



(a) Initial IF-STP **(b)** Optimized IF-STP

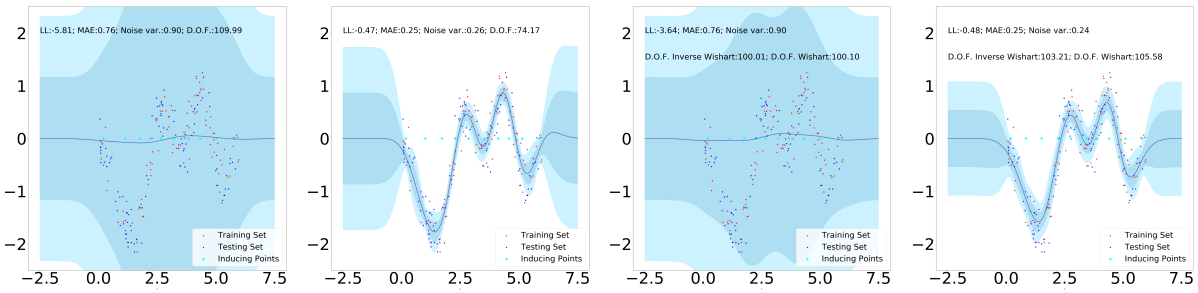
(c) Initial IF-CSTP **(d)** Optimized IF-CSTP

Figure 5.4: Inverse free models correctly recover optimal inducing points' locations. Predictive mean and variance plots at initialization and after optimization, split in parametric and non-parametric components. Inducing points' location is initialized to be outside the training set manifold.

in situations when the inducing points' locations Z do no suffer large shifts (such as is the case with the "banana" dataset for 32 and 64 inducing points)

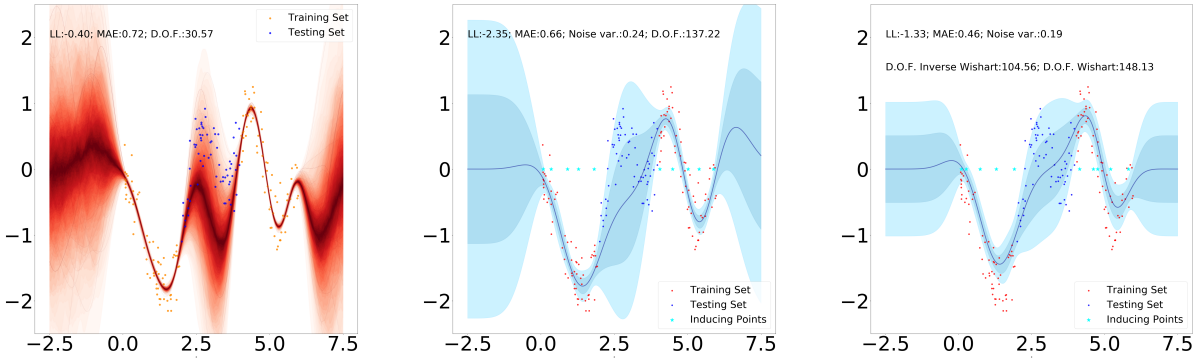
The lower bounds manage to recover accurate estimates of T for IF-CSTP, respectively Σ_{uu} and $\Sigma_{ff\cdot u}$ for IF-STP (Figure 5.5).

Reliability of uncertainty estimates. *We are interested to stress test our proposed model in three different scenarios, which will individually point towards biases or discrepancies in the final estimates for $\Sigma_{ff\cdot u}$ and $\Sigma_{fu}\Sigma_{uu}^{-1}$. For this we use the "snelson" dataset, with the training set taken to comprise the intervals between 0.0 and 2.0, respectively 4.0 and 6.5. Thereby, in an ideal scenario we would expect our model to offer high uncertainty estimates between 2.0 and 4.0. From Figure 5.6 we can*



(a) Initial IF-STP **(b)** Optimized IF-STP **(c)** Initial IF-CSTP **(d)** Optimized IF-CSTP

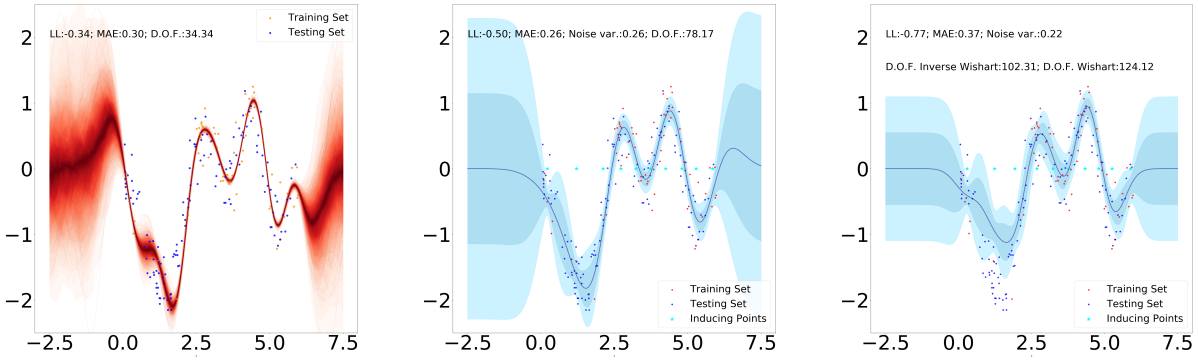
Figure 5.5: Correct optimization of variational parameters. Predictive mean and variance plots at initialization and after optimization. Variational parameters are erroneously initialized.



(a) Student-t Processes **(b)** IF-STP **(c)** IF-CSTP

Figure 5.6: Reliability of *in-between* uncertainty estimates. Predictive mean and distributional variance for models trained on "snelson with gap" dataset. Likelihood variance is not added.

notice that indeed both IF-STP and IF-CSTP do not suffer from any pathologies.



(a) Student-t Processes **(b)** IF-STP **(c)** IF-CSTP

Figure 5.7: Reliability of epistemic uncertainty estimates. Predictive mean and distributional variance for models trained on locally subsampled “snelson” dataset. Likelihood variance is not added.

Subsampling the “snelson” dataset allows us to discern if our method is capable to increase its uncertainty. For this we only subsample data between 0.0 and 2.5. Naturally, one should expect an increase in uncertainty just in that interval, with the remainder of the data range reverting back

to levels seen for the full "snelson" dataset. We can notice this increase in uncertainty within the subsampled interval, with a relative underfitting of IF-CSTP (Figure 5.7).

Results on medium-scale datasets. *We evaluate our proposed model (IF-CSTP) on selected UCI datasets. We attempted to train IF-CSTP-NGD under similar training conditions. However, the training was unstable to the point of non-convergence so we did not include results here. We defer results on the NGD variant to the next paragraph.*

The gap between inverse free ELBOs and the SVGP ELBO is predominantly caused by the KL divergence pertaining to T in the case of IF-CSTP, respectively the KL divergence associated to U for IF-ReSVGP. From Figure 5.8 we can notice that in terms of normalized training set log likelihood IF-CSTP and IF-ReSVGP are similar in behaviour to SVGP. IF-CSTP's testing time behaviour almost approaches SVGP with a noticeable example on "Concrete". Conversely, on "elevators" IF-CSTP is underfitting. In some cases, such as on "Protein", the test time behaviour of IF-CSTP is similar to SVGP except for 250 inducing points, where the gap widens considerably. This might indicate some pathological behaviour of IF-CSTP when applied on large-scale datasets that require a large number of inducing points.

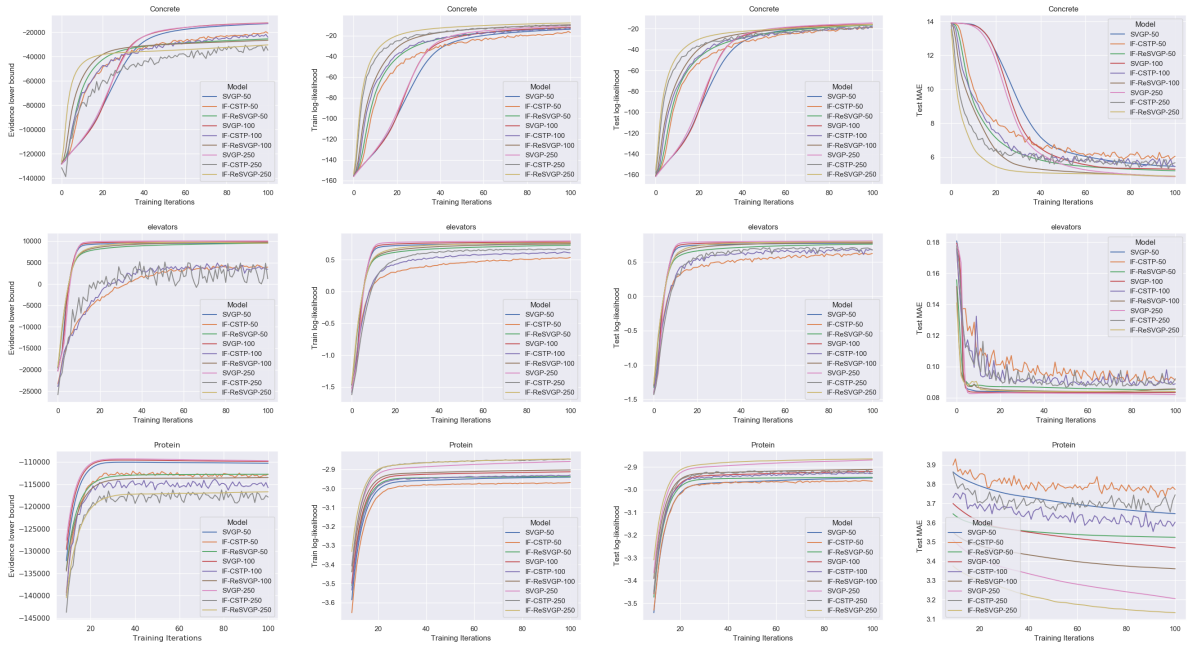


Figure 5.8: UCI results. Evidence lower bound, normalized training and testing log-likelihoods and mean absolute error values during optimization for varying inducing points numbers.

We can notice that our proposed model is significantly more noisy during optimization in comparison to IF-ReSVGP due to sampling across three different random variables ($\Sigma_{fu}\Sigma_{uu}^{-1}$, $\Sigma_{ff\cdot u}$ and T). An important aspect in the ELBO for IF-CSTP is the lower bound on $\log|K_{uu}|$ present in the KL divergence of T. This lower bound involves an iterative PCG routine (see Proposition 5.4.1) which depending on the condition number will either take more or less iterations. The aim is to lower the

number of iterations required by the PCG routine so as to speed inference. From Figure 5.9 we can see that for 250 dimensional matrices the number of CG iteration is only a fraction of that number, which slowly decreases during optimization as the preconditioning matrix $\tilde{K}_{uu}^{-1}$ gets better at approximating the real inverse.

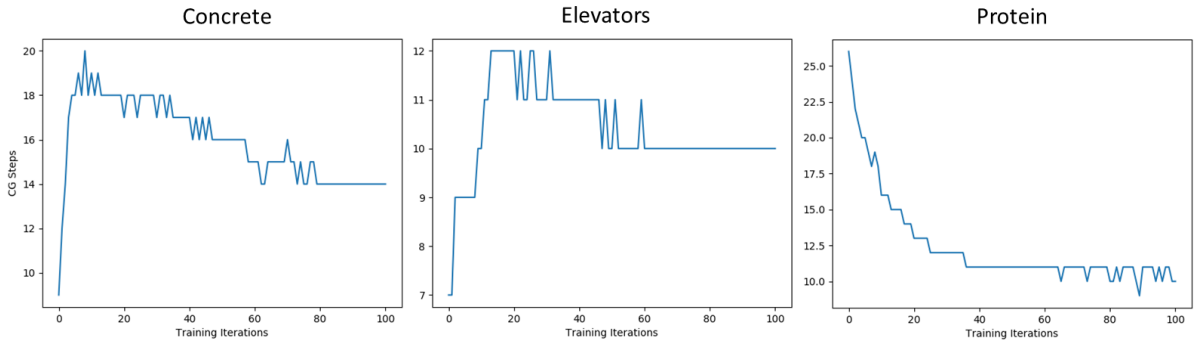


Figure 5.9: Effect of approximate posterior T on condition number. Number of CG steps needed to satisfy a tight tolerance for IF-CSTP trained on various datasets with 250 inducing points.

Natural Gradients variant converges in GP scenario. *IF-CSTP-NGD converges during optimization only in the case that we set $\nu_{f,q}^{W^{-1}} \to \infty$, which corresponds to equations* (5.77) *and* (5.78) *acting akin to a GP. We rebrand this variant as IF-GP-NGD.*

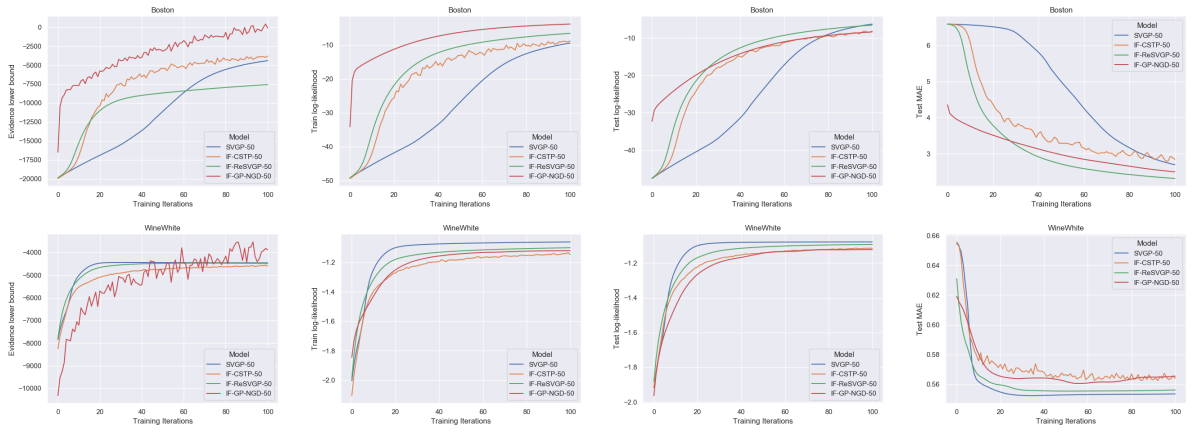


Figure 5.10: Natural Gradient variant results. Evidence lower bound, normalized training and testing log-likelihood and mean absolute error values during optimization.

From Figure 5.10 we can see that IF-GP-NGD better fits the training data better compared to IF-CSTP by looking at the normalized training set log likelihood, with roughly similar behaviour at testing time. Explicitly computing the ELBO is not required for the NGD update rules for U and T. However, it is required to obtain gradients for hyperparameters. Small biases introduced by the PCG routine can have a major effect on the estimated ELBO, such as is the case with the "WineWhite" dataset, where the KL divergence for T erroneously becomes negative, thus resulting in an over-estimation of the real ELBO. This pathology will motivate further research into using bias-free PCG routines or removing altogether the need to estimate the ELBO for hyperparameter gradients.

5.7 Discussion

We commence this section by positioning our proposed method in existing literature, drawing parallels and highlighting differences with respect to prior work. Subsequently, we conclude this chapter with a set of remarks and final thoughts.

5.7.1 Relation to prior work

In this subsection we consider similarities and differences between our proposed models in this chapter and already existing ones in literature, firstly from the point of view of inverse-free GP, followed by sparse formulations of TP.

Inverse-free models. *We now consider possible similarities between our proposed scalable model IF-CSTP and the inverse-free model introduced in Wilk et al. (2020b) (see subsection 2.3 for technical details). We commence with IF-ReSVGP, which had the following ELBO:*

$$\mathcal{L}_{IF-ReSVGP} = \sum_{i=1}^{N} \mathbb{E}_{\mathcal{N}\left(f_i|\hat{U}(x_i),\hat{\Sigma}(x_i)\right)} \left[\log p(y_i \mid f(x_i))\right] - \mathrm{KL}\left[\mathcal{N}\left(m_U^*, S_U^*\right) \| \mathcal{N}\left(0, K_{uu}\right)\right] \tag{5.111}$$

$$\hat{U}(x_i) = K_{f_i u}\hat{u} \tag{5.112}$$

$$\hat{\Sigma}(x_i) = K_{f_i f_i} + K_{f_i u}TK_{uu}TK_{uf_i} - 2K_{f_i u}TK_{uf_i} + K_{f_n u}\hat{S}K_{uf_n}\ , \tag{5.113}$$

where we remind ourselves that the authors have performed the following reparametrizations: $K_{uu}^{-1}m_U = \hat{u}$ and $K_{uu}^{-1}S_U K_{uu}^{-1} = \hat{S}$ and in the upper bound on the non-parametric variance T essentially acts as a point-estimate of K_{uu}^{-1}. With this in mind, we can take the following posterior $q(\hat{U}) = \mathcal{N}\left(Tm_U, TS_U T\right)$, for which we can update the moments of the predictive equation as follows:

$$\hat{U}(x_i) = K_{f_i u}Tm_U \tag{5.114}$$

$$\hat{\Sigma}(x_i) = K_{f_i f_i} + K_{f_i u}TK_{uu}TK_{uf_i} - 2K_{f_i u}TK_{uf_i} + K_{f_i u}TS_U TK_{uf_i}\ . \tag{5.115}$$

We can now see that this corresponds in form to the posterior predictive equations of IF-CSTP when $\nu_{f,q}^{W^{-1}} \to \infty$. Therefore, in the data fit term part of their ELBOs, the two models are identical in the given scenario.

To make the analogy even further, one could lower bound $L_{IF-ReSVGP}$ by imposing a prior over T

in the following generative process of the posterior:

$$T \sim \mathcal{W}_m \left(\nu_{f,p}^W,\ \frac{1}{\nu_{f,p}^W} \tilde{K}_{uu}^{-1} \right) \tag{5.116}$$

$$U \sim \mathcal{N}\left(m_U, S_U\right) \tag{5.117}$$

$$f \mid U, T \sim \mathcal{N}\left(K_{fu}TU, K_{ff} + K_{fu}TK_{uu}TK_{uf} - 2K_{fu}TK_{uf}\right)\ , \tag{5.118}$$

where we remind ourselves that in the original formulation of the model, T took the role of a point estimate of K_{uu}^{-1}, hence the rationale behind now imposing a probabilistic perspective over T as a realization from a Wishart distribution. Following standard VI framework we can obtain the following ELBO for the Bayesian variant of IF-ReSVGP:

$$\begin{aligned} \mathcal{L}_{IF-BReSVGP} = \mathbb{E}_{q(T)} \left[\sum_{i=1}^{N} \mathbb{E}_{\mathcal{N}\left(f_i \mid \hat{U}(x_i), \hat{\Sigma}(x_i)\right)} \left[\log p(y_i \mid f(x_i)) \right] \right] \\ - \mathbb{E}_{q(T)} \left[\mathrm{KL}\left[\mathcal{N}\left(u^*, S^*\right) \| \mathcal{N}\left(0, K_{uu}\right) \right] \right] \\ - \mathrm{KL}\left[\mathcal{W}_m^{-1}\left(\nu_{f,q}^{W^{-1}}, \tilde{K}_{uu}^{-1}\right) \| \mathcal{W}_m^{-1}\left(\nu_{f,p}^{W^{-1}}, K_{uu}^{-1}\right) \right]\ . \end{aligned} \tag{5.119}$$

The equivalence between the two models in the data fit term and the KL divergence pertaining to T is obvious if we inspect again $L_{IF-CSTP}$ as defined in equation (5.72)*. However, we are left to investigate the correspondence in the KL divergence associated to inducing point values, which we now expand:*

$$\mathbb{E}_{q(T)} \left[\frac{1}{2} \left[\log \left| K_{uu} \right| - \log \left| S^* \right| - d + \mathrm{Tr}\left[K_{uu}^{-1} S^* \right] + u^{*\top} K_{uu}^{-1} u^* \right] \right]\ . \tag{5.120}$$

For ease of comparison, we reiterate the KL divergence for inducing point values for IF-CSTP (in the scenario $\nu_{f,q}^{W^{-1}} \to \infty$):

$$\mathbb{E}_{q(T)} \left[\frac{1}{2} \left[\log \left| T^{-1} \right| - \log \left| S_U \right| - d + \mathrm{Tr}\left[TS \right] + m_U^\top T m_U \right] \right]\ . \tag{5.121}$$

A visible difference between IF-BReSVGP and IF-CSTP (in the case that $\nu_{f,q}^{W^{-1}} \to \infty$) resides in the highly structured posterior $q(S^*)$ (from IF-BReSVGP) compared to the completely unstructured posterior $q(S)$ (from IF-CSTP). This is caused by the fact that the upper bound on non-parametric variance is imposed through the posterior as opposed to IF-CSTP, where the upper bound is imposed through the prior. To better see this difference, we remind ourselves the chosen conditional distribution in the augmented space of covariance matrices of IF-CSTP:

$$\Sigma_{fu,fu} \mid T \sim \mathcal{W}_{n+m}^{-1} \left(\nu_{f,p}^{W^{-1}} + n,\ \nu_{f,p}^{W^{-1}} \begin{bmatrix} T^{-1} & K_{uf} \\ K_{fu} & \overline{K_{ff \cdot u}} + K_{fu}TK_{uf} \end{bmatrix} \right)\ . \tag{5.122}$$

Hence, the upper bound on non-parametric variance is already imposed in the prior over $\Sigma_{fu,fu}$ given T. Besides that, the elimination of matrix inverses in the parametric part of IF-BReSVGP is achieved again through the explicit construction of the posterior. Conversely, in IF-CSTP this is achieved through the explicit link between T and Σ_{uu}^{-1}, a bond which does not exist in IF-BReSVGP, since T is considered a general random variable that is incorporated in the posterior of $\hat{U}$.

Sparse formulations of Student-t Processes. *Due to the generally perceived problems regarding their mathematical tractability in comparison to GP, TP have received less attention since the seminal paper of Shah et al. (2014). More specifically, advances in the scalability of GP (Hensman et al. 2013; Titsias 2009) or inference methods for non-conjugate likelihoods (Hensman et al. 2015; Khan et al. 2012; Shang and Chan 2013; Sheth et al. 2015) have not necessarily been translated to TP. In Roetzer (2020), the author proposes to solve these current drawbacks by extending these advances to TP using the concept of t-relaxation, which can be defined as "A reformulation of a problem or parts of a problem in such a way that the original problem is recovered by taking the limit as t approaches 1.". To put things into perspective, two common examples of t-relaxation are the t-exponential (e.g., used to create the family of t-exponential distributions (Tsallis 1988)) and the t-logarithm (e.g., used in Tsallis entropy (Tsallis and Brigatti 2004)), which are generalizations of their standard versions. These two functions have also found a use in variational inference methods for the family of t-exponential distributions (Ding et al. 2011; Futami et al. 2017), which constitutes the theoretical backbone of methods developed in Roetzer (2020). In his PhD thesis, Roetzer proposes a generalization of the approach introduced in Hensman et al. (2015), by using a t-relaxed version of the ELBO, which results in a scalable sparse version of TP, where the posteriors $q(F)$ and $q(U)$ both follow multivariate Student-t distributions. Their resulting ELBO consists of a t-relaxed expected log likelihood term (which converges to the standard case when $t \to 1$, in which case it would equate to our proposed formulation) and a t-divergence (Ding et al. 2011) between $q(U)$ and $p(U)$, which has an analytical expression for multivariate Student-t distributions, unlike the case for KL divergences. Conversely, due to the lack of analytical formulas for KL divergences between multivariate Student-t distributions, we parametrized our posterior $q(U)$ as a multivariate normal distribution, as opposed to a multivariate Student-t distribution. As for predictive equations, the two formulations result in similar formulas at testing time. In Roetzer (2020), the author states that their proposed sparse formulation of TP using t-relaxation fails to converge on real datasets. Hence, doing a comparison between their proposed method (alongside improvements to it) and our proposed method remains a timely prospective research topic, one that could result in STP having a wider usage by the community.*

5.7.2 Concluding remarks

Using our matrix inversion-free variational lower bound we showed similar behaviour to matrix inversion-dependent counterparts on toy tasks, alongside approximately similar behaviour on real-life datasets.

However, scaling our proposed method to work on large-scale data still remains a research avenue as there is a tendency of over-stability of Z for large numbers of inducing points due to the large penalty stemming from $\mathrm{KL}\left[q(T)\|p(T)\right]$. Expanding on this, from equation (5.82) *we can see that the terms $\frac{\nu_{f,q}^W}{2}\left[\log|K_{uu}| + \log\left|\tilde{K}_{uu}^{-1}\right|\right]$ and $\frac{\nu_{f,q}^W}{2}\left[\mathrm{Tr}\left[\frac{\nu_p}{\nu_q}K_{uu}\tilde{K}_{uu}^{-1}\right] - m\right]$ are scaled by $\nu_{f,q}^W = \nu_q^W + m + 1$, where $\nu_q^W \geq 0$. Hence, any difference between K_{uu}^{-1} and $\tilde{K}_{uu}^{-1}$ is further scaled by a scalar which has a minimum value equal to the number of inducing points. One can rapidly see that this might constitute an issue when using a large number of inducing points, as any gradient update for kernel hyperparameters or Z will incur a large negative penalty from this KL divergence. We illustrated this drawback in our experiments on the "banana" dataset. What was perhaps surprising in the aforementioned experiment was that IF-ReSVGP did not exhibit similar optimization behaviour for Z as in the case of SVGP even though this particular model does not have a regularization penalty on T as our proposed models. This lack of a penalty term as given for example by $\mathrm{KL}\left[q(T)\|p(T)\right]$ means that in that case T is not explicitly encouraged towards K_{uu}^{-1}. Hence, we can posit that for IF-ReSVGP if T doesn't rapidly find the real solution K_{uu}^{-1} given a new set of hyperparameter and Z values, it will not exhibit a similar behaviour to SVGP. Conversely, IF-CSTP explicitly guide T towards K_{uu}^{-1} via the KL divergence. However it is precisely due to this regularization term that kernel hyperparameters or Z might not change due to possibly high penalty.*

Whereas we have introduced generative processes based on Student-t Processes, what we have noticed in practice is that both $\nu_q^{W^{-1}}$ and ν_q^W will increase during optimization, thus any model will slowly converge towards an inverse-free GP. We have explored various initialization schemes for $\nu_q^{W^{-1}}$ and ν_q^W and what has always worked in practice is to initialize them by using a softplus function using the number of inducing points as the function argument. For models with small number of inducing points, if one initializes the aforementioned starting from 0, the resulting model would fail to converge. This observation falls in line with previous research (Jylänki et al. 2011), which also highlighted that performing inference with low degrees of freedom results in unstable behaviour. With this in mind, in our experiments with the NGD variant even upon starting with high degrees of freedom at initialization, this did not ensure convergence. Hence, we were only able to use the NGD variant of our proposed model in the limiting case of behaving as a GP, showing promising initial results compared to the standard gradients variant. Perhaps the greatest advantage of IF-CSTP-NGD is that its variational parameter natural gradients do not involve any computationally expensive operations such as matrix inversion or computing log determinants as in the explicit computation of the ELBO.

Setting a low tolerance for the CG-based lower bound on log determinant does not guarantee that our objective remains a lower bound. Nevertheless, we have shown empirically that this does not necessarily constitute a problem in practice, as our preconditioned CG routine generally has a low condition number, thus ensuring the convergence of the PCG routine in only a couple of iterations despite operating on high-dimensional matrices. This constitutes an advantage of our proposed Bayesian hierarchical model as there is a direct link between T and K_{uu}^{-1}, which is used as the pre-

conditioner. This is in stark contrast to preconditioning used in other inverse-free GP models from literature (Wang et al. 2019; Wilk et al. 2020b). Nevertheless, in our experiments we did encounter one instance when due to biases associated to the CG-based lower bound our ELBO did not represent a lower bound anymore (see "WineWhite" experiment in Figure 5.10). With this in mind, bias-free CG algorithms are required to guard against this pathology. An alternative method is to eliminate the need for computing the ELBO to obtain hyperparameter gradients, leaving this as a prospective research avenue.

Lastly, a drawback of our current work is the lack of competitiveness in terms of speed with SVGP. To obtain a lower bound on log determinants in our ELBO, we make use of Proposition 5.4.1, which involved PCG. In this chapter we have used the standard implementation of PCG, which was shown to not be optimal on GPU hardware (Gardner et al. 2018b). Moreover, truncated conjugate gradients have been proposed (Filippone and Engler 2015b; Potapczynski et al. 2021) to provide bias-free solutions. These recent advancements should be used in future iterations of this chapter to ensure faster and more accurate inference, so as to make our proposed model competitive in inference speed.

Chapter 6

Conclusions

In this chapter we revisit our research aims set out in the introduction and see whether we have achieved them in our contribution chapters. Subsequently, we discuss limitations and avenues for future research.

6.1 Main Contributions

At the beginning of the thesis in Section 1.4 we laid out our research goals, which we recall now.

Intended Research Aims

- ***A*** *Correctly propagating distributional uncertainty in Hierarchical Gaussian Processes*
- ***B*** Distance-aware *GP-based architectures in medical imaging tasks*
- ***C*** *Matrix inversion-free inference in sparse Student-t Processes*

In this section we summarize the main contributions and conclusions of each chapter.

Chapter *3 explored whether distributional uncertainty is reliably propagated in DGP. We provided both theoretical and empirical evidence that this does not necessarily occur in practice nor in some realistic case studies. The main contribution of this chapter was the introduction of a variant of DGP which operates on both Euclidean and Wasserstein space. We showed again both empirically and theoretically that this probabilistic re-formulation of DGP is more reliable in propagating distributional uncertainty, or more specifically what was initially identified as an outlier by layers lower in the hierarchy should still be identified as an outlier in layers higher in the hierarchy. We hope that this line of research will inspire future work on using DDGP as the building blocks for OOD detection models.*

Using DistGP as the building blocks for OOD detection in medical imaging tasks was the main focus in ***Chapter*** *4. The challenge was in constructing a module that can be easily extended to convolutional architectures specific to classification or segmentation, all while still maintaining OOD detection capabilities. We proposed a drop-in replacement for standard convolutional layers, whereby we impose Lipschitz continuity on the affine operators so as to satisfy* distance-awareness *conditions, respectively using DistGP as "activation functions" to introduce non-linearities. DistGP are by definition* distance-aware*. We demonstrated on both image classification and medical imaging segmentation tasks that our proposed model provides better OOD detection results than state-of-the-art "OOD detection models" from literature. One drawback of using DistGP-Seg for image segmentation is the relative lag in performance compared to a U-Net.*

A visible impediment towards the wider adoption of Distributional Gaussian Processes Layers as the building blocks of predictive models is related to the high computation costs associated to matrix inversion operations and log determinants in the evidence lower bound. One can easily see that for deep networks, these costs would inhibit practitioners from using DistGP Layers as the training time would be orders of magnitude higher compared to a DNN. Thus, eliminating these unwanted computational operations was the primary objective of ***Chapter*** *5, where we proposed a matrix inversion-free lower bound on conditional sparse Student-t Processes. Log determinant operations are avoided*

through the use of an upper bound involving preconditioned conjugate gradients. Through experiments on both toy and real-life datasets, we have shown that our proposed inverse-free sparse Student-t Process almost approaches the behaviour of SVGP. Despite several limitations and pending future work, which we will outline subsequently, we are confident that this chapter will constitute a stepping stone in future research work on inverse-free GP.

6.2 Limitations & Future Research

We conclude this thesis with a discussion on some limitations of our contribution chapters, alongside suggestions of prospective research avenues to consider in the future to address these limitations.

6.2.1 Uncertainty Decomposition & DGP variants

This thesis has been developed under the assumption that SVGP (Hensman et al. 2013) represents the best possible variational approximation of a full GP. Indeed, if we consider comparison papers (Bauer et al. 2016), SVGP exhibit a set of advantages over previous sparse approximations (see subsection 2.2.2 for details). However, in recent years SVGP has been challenged from a diverse set of view points. First, Jankowiak et al. (2020b) noticed that the non-input dependent uncertainty σ^2_{noise} (i.e., aleatoric uncertainty) and the input dependent uncertainty $\tilde{\Sigma}(x)$ (i.e., epistemic & distributional uncertainty, as defined in equation (2.42)) are present in the SVGP ELBO in an asymmetrical manner (i.e., $\tilde{\Sigma}(x)$ does not appear in the data fit term that results from the expected log likelihood $\mathbb{E}_{q(u)}\left[\log p(y|x,u)\right]$ in equation (2.43)), whereas at testing time these two sources of uncertainty are present in an additive fashion (i.e., $p(y^* \mid x^*) = \mathcal{N}\left(y^* \mid \tilde{U}(x^*),\, \tilde{\Sigma}(x^*) + \sigma^2_{noise}\right)$, where $\tilde{U}(x^*)$ is defined in equation (2.41)). The authors argue that this discrepancy in how uncertainties are present at training and testing time is the root cause behind why in SVGP the resulting predictive distributions exhibit underestimated variance. The authors emphasize that in regression tasks the overall variance is dominated by aleatoric uncertainty, making little use of input dependent uncertainty. To combat this pathology, the authors propose the Parametric Gaussian Process Regressor (PGPR), which boils down to devising an alternative ELBO that incorporates input and non-input dependent uncertainty in a symmetric manner in the ELBO, more precisely they propose the following: $\mathcal{L}_{PGPR} = \sum_{i=1}^{N} \log \mathcal{N}\left(y_i \mid \tilde{U}(x), \tilde{\Sigma}(x) + \sigma^2_{noise}\right) - \beta_{reg}\,\mathrm{KL}\left[q(U)\|p(U)\right]$, where β_{reg} is a hyperparameter that controls the amount of regularization. In Jankowiak et al. (2020a) the authors propose extending the framework of PGPR to DGP, developing a model coined Deep Sigma Point Processes which exhibits enhanced calibration in predictive distributions compared to DGP alongside the advantage of not having to sample intermediate latent functions in the ELBO, instead using a quadrature scheme. Some other variants of DGP are the Robust DGP (Knoblauch 2019), Conditional DGP (Lu and Shafto 2021) and Deep Kernel Processes (Aitchison et al. 2020). A potentially interesting research avenue is to explore how uncertainty is propagated in these DGP variants and whether there are any

differences with respect to our results from Chapter 3. These aforementioned DGP variants utilize the framework of inducing points to perform approximate inference in the hierarchy, which as we have seen in Chapter 3 can have deleterious effects on OOD detection capabilities. Conversely, Random Feature Expansions for DGP (Cutajar et al. 2017) proposes an alternative scalable approximation of DGP, devising a framework that does not involve any matrix decompositions based on the number of inducing points. Briefly, this model approximates the GP present in hidden layers using random feature expansions (Rahimi and Recht 2007). Seeing whether this inducing points-free approximate inference method for DGP suffers from the same pathologies discovered in Chapter 3 remains a prospective research topic.

Besides considering uncertainty propagation in variations of DGP, another important aspect is how to decompose the total uncertainty in the model so as to identify root causes of uncertainty. Throughout the thesis, we have employed the uncertainty decomposition framework introduced Malinin and Gales (2018) where total uncertainty is split between aleatoric, epistemic and distributional uncertainty. However, there are alternative decomposition frameworks such as the Bayesian nonparametric ensemble (BNE) framework proposed in Liu et al. (2019), which splits total uncertainty into aleatoric (irreducible due to stochastic variability inherent in the data generating process) and epistemic uncertainty (lack of knowledge about data generating mechanism). The authors argue that the latter uncertainty can be further divided into parametric uncertainty (uncertainty in estimating model parameters under current model specification) which is akin to the parametric variance in SVGP, respectively structural uncertainty (uncertainty whether model specification is sufficient for describing the data). The BNE framework proposes that structural uncertainty is caused by biases pertaining to the limited hypothesis space of an ensemble, respectively biases caused in the mismatch between the true data distribution (can be multi-modal, non-symmetric or heavy-tailed) and the usually assumed Gaussian additive noise. The former bias results in an uncertainty which grows as test points are further away from training set manifold, essentially acting in a similar fashion to distributional uncertainty. The additional uncertainty in the BNE framework compared to our assumed framework is related to the uncertainty in the mismatch between our Gaussian distributed predictions and the true data generating process. A possible solution to eliminate uncertainties pertaining to this distribution mismatch is to use an alternative to GP, which allow for more generalized distributions. Benavoli et al. (2020) have assumed that the data follows a skew-normal multivariate normal (Azzalini 2013), respectively Shah et al. (2014) assumed a generative process reliant on the multivariate Student's t distribution. A potential research avenue is to unify these two approaches, hence obtaining posterior latent function that exhibit both skewness and heavy-tails. This could be achieved by making use of the multivariate extended skew-t distribution (Arellano-Valle and Genton 2010). An orthogonal approach would be to use Normalizing Flows (NF) (Rezende and Mohamed 2015) on top of a GP (Maroñas et al. 2021), which due to the ability of NF to model implicit distributions would guarantee that there is no bias due to assumed predictive distribution.

6.2.2 *Enhanced Approximate Inference for* really deep *DGP*

In Chapter 4 we have noticed that using DSVI (Salimbeni and Deisenroth 2017) for models with 18 layers results in reduced performance in comparison to counterparts with just two or three layers. We posit that the reduction in performance is caused by the DSVI framework which is not capable to scale to deep networks with possibly complex multi-modal and correlated posteriors between layers, therefore to scale DistGP Layers to modern architectures we need to find alternative approximate inference frameworks. Indeed, it is common knowledge that one of the biggest challenges with DGP (variants included) is that this class of models does not allow for exact inference. Throughout this thesis, we have employed the DSVI framework proposed in Salimbeni and Deisenroth (2017), which the authors have shown to outperform DGP inference frameworks based on Expectation Propagation (Bui et al. 2016). DSVI uses a factorized posterior across layers, assigning Gaussian distributions layer-wise to inducing point values. One can immediately see how this approximate posterior can fail in the case that the real posterior is multi-modal (e.g., multi-modal toy problem experiment in Havasi et al. (2018)). To address this modelling drawback, Havasi et al. (2018) have proposed the Moving Window Monte Carlo Expectation Maximization (MWMCEM) framework for obtaining multi-modal posteriors in DGP, showing improved performance at a reduced computational cost compared to DSVI. As opposed to using MCMC methods to obtain complex and multi-modal posteriors, implicit variational inference (IVI) (Huszár 2017) has been used to train DGP (Yu et al. 2019) or a structured variant of semi-implicit variational inference (SIVI) (Yin and Zhou 2018) being used as in Molchanova et al. (2019). With this in mind, one possible way of arriving at complex multi-modal posteriors which are not factorized between layers is to use a hypernetwork (Ha et al. 2016) to generate inducing point values across layers within the inference framework of more recent variants of SIVI (Molchanov et al. 2019; Sobolev and Vetrov 2019). Alternatively, another possible formulation of DGP with correlations between layers is the "global" inducing points approach introduced in Ober and Aitchison (2021). Seeing if these advances in approximate inference for DGP would translate to increased performance in deep convolutional architectures based on the DistGP Layers module remains a research avenue to be explored.

6.2.3 *Refinements on inverse-free Student-t Processes*

Optimization of variational parameters and hyperparameters. *The simultaneous joint optimization of both variational parameters (U and T) and hyperparameters $\theta = \{Z, \sigma^2, l^2\}$ in IF-CSTP might not be the optimal route to take, as for example minor changes to θ can cause massive differences in K_{uu}^{-1}, which T is attempting to approximate. Hence, during optimization the gradients pertaining to T are playing a "catch-up game" to keep up with the trajectories of hyperparameters. We posit that a different training scheme might be suited for our scenario. Recently, Adam et al. (2021) introduced a Variational Expectation-Maximization (VEM) scheme for training SVGP, which involves an alternative*

optimization between variational parameters and hyperparameters at the t-th iteration as follows:

$$\textit{E-step}: \ \xi_t^* \leftarrow \arg\max_{\xi} \mathcal{L}_{IF-CSTP}\left(\xi, \theta_t\right) \tag{6.1}$$

$$\textit{M-step}: \ \theta_{t+1}^* \leftarrow \arg\max_{\theta} \mathcal{L}_{IF-CSTP}\left(\xi_t^*, \theta\right) \ , \tag{6.2}$$

where for both the E-step and M-step we iteratively find the optimum by standard gradient ascent, such as for the E-step: $\xi_t^{k+1} \leftarrow \xi_t^k + \beta_t \nabla_\xi \mathcal{L}_{IF-CSTP}\left(\xi_t^k, \theta_t\right)$, where β_t is the learning rate and we optimize for a pre-defined K numbers of steps. The intuition is that during the E-step, the variational parameters ξ associated to $q(T)$ will be able to stably converge towards the real solution K_{uu}^{-1} since under this framework it doesn't have to converge to a moving target.

Optimizing hyperparameters in NGD case. *Whereas in section 5.5 we provided an inverse-free framework for computing NGD updates for variational parameters (i.e., U and T for IF-CSTP) that does not involve iterative updates from CG routines, we omitted how to take GD updates for kernel hyperparameters and Z (will be denoted as θ). The reason is that computing these derivatives would involve matrix inversion operations, more specifically we are interested in computing the following:*

$$\nabla_\theta \int q(T) \log \frac{q(T)}{p(K_{uu}^{-1})} \, \mathrm{d}T = \frac{\partial - \frac{\nu_p}{2} \log \left|\nu_p K_{uu}\right| + \frac{\nu_q}{2} \operatorname{Tr}\left[\frac{\nu_p}{\nu_q} K_{uu} \tilde{K}_{uu}^{-1}\right]}{\partial \theta} \tag{6.3}$$

$$= -\frac{\nu_p}{2} \operatorname{Tr}\left[\left(\nu_p K_{uu}\right)^{-1} \frac{\partial \nu_p K_{uu}}{\partial \theta}\right] + \frac{\partial \frac{\nu_q}{2} \operatorname{Tr}\left[\frac{\nu_p}{\nu_q} K_{uu} \tilde{K}_{uu}^{-1}\right]}{\partial \theta} \ , \tag{6.4}$$

where we only kept terms of $\mathrm{KL}\left[q(T) \| p(T)\right]$ (see equation (5.82)) that involve θ. We notice that we now need to compute K_{uu}^{-1} to obtain gradients, which is a computational cost we do not intend to incur. Hence, applying Proposition 5.4.1 is required again so as to eliminate matrix inversion, with the drawback of increased computational cost in the case that our variational distribution of T diverges from the real solution to K_{uu}^{-1}. How to efficiently take gradients with respect to kernel hyperparameters still remains a future research avenue.

Meta-learning variational distribution associated to T. *In Chapter 5, we have defined our approximate posterior over T through a Cholesky decomposition, keeping in line with previous approaches to VI in SGP (Hensman et al. 2013; Salimbeni et al. 2018). However, in this case there is an intrinsic link between T and K_{uu}^{-1}. Hence, it would make sense to exploit this natural link by devising an approximate posterior $q(T)$ that makes use of kernel hyperparameters and inducing points' locations Z (i.e., θ). Therefore, $L_{\tilde{K}_{uu}^{-1}}$ can be parametrized through a neural network that is fed θ as the input. This hypernetwork tasked to generate $L_{\tilde{K}_{uu}^{-1}}$ can be trained from scratch on each dataset or alternatively, one could draw inspiration from Liu et al. (2020b), where authors propose a specialized self-attention network that is fed datasets and outputs hyperparameters. In our case, this permuta-*

tion invariant network would have hyperparameters as input and it would produce $L_{\tilde{K}_{uu}^{-1}}$. Due to the self-attention module's invariance to dimensions and data (in this case Z), pre-training this architecture on a set of representative datasets can provide an accurate starting point, which could get fine-tuned during training of our inverse-free model. This remains a prospective research topic.

Faster and unbiased Conjugate Gradients & Gradient Variance Reduction *A major drawback of our current work is the lack of competitiveness in terms of speed with SVGP. To obtain a lower bound on log determinants in our ELBO, we make use of Proposition 5.4.1 which involved PCG. In this chapter we have used the standard implementation of PCG, which was shown to not be optimal on GPU hardware (Gardner et al. 2018b). Moreover, truncated conjugate gradients have been proposed (Filippone and Engler 2015b; Potapczynski et al. 2021) to provide bias-free solutions. These recent advancements should be used in future iterations of this chapter to ensure faster and more accurate inference, so as to make our proposed model competitive in inference speed.*

Lastly, both standard and natural gradients versions of our proposed model suffer from noisy gradients caused by having to sample from a Bayesian hierarchical generative process, more precisely from T, $\Sigma_{fu}\Sigma_{uu}^{-1}$ and $\Sigma_{ff\cdot u}$. In the case of NGD this increased stochasticity negatively affects training to the point that it only works in the limiting case that we need to impose a GP behaviour as opposed to a Student-T Process to enable convergence. Marginalizing out the different stochastic components of our generative process is analytically intractable. However, we can use approximate moments instead. We can re-express our prior generative process as follows:

$$p\left(T\right) = \mathcal{W}_m\left(\nu_{f,q}^{W},\ \frac{1}{\nu_{f,q}^{W}}K_{uu}^{-1}\right) \tag{6.5}$$

$$p\left(f, U \mid T\right) = \mathcal{MVT}\left(\nu_{f,q}^{W^{-1}} + n,\ 0,\ \nu_{f,q}^{W^{-1}}\begin{bmatrix} T^{-1} & K_{uf} \\ K_{fu} & \overline{K_{ff\cdot u}} + K_{fu}TK_{uf}\end{bmatrix}\right). \tag{6.6}$$

Making use of conditional multivariate-t identities (Shah et al. 2014) we have:

$$p\left(f \mid U, T\right) \sim \mathcal{MVT}\left(\nu_{f,q}^{W^{-1}} + n + m, K_{fu}TU, \frac{\nu_{f,q}^{W^{-1}} + n + U^{\top}TU - 2}{\nu_{f,q}^{W^{-1}} + n + m - 2}\overline{K_{ff\cdot u}}\right). \tag{6.7}$$

Using the law of iterated expectations and the law of conditional variance we obtain the following first two moments for $q(f \mid T) = \int p(f \mid U, T)q(U)\,\mathrm{d}U$:

$$\mathbb{E}\left[f \mid T\right] = \mathbb{E}_{q(U)}\left[\mathbb{E}\left[f \mid U, T\right]\right] = K_{fu}Tm_U \tag{6.8}$$

$$\mathbb{V}[f \mid T] = \mathbb{E}_{q(U)}[\mathbb{V}[f \mid U, T]] + \mathbb{V}_{q(U)}[\mathbb{E}[f \mid U, T]] \tag{6.9}$$

$$= \frac{\nu_{f,q}^{W^{-1}} + n + \mathrm{Tr}\left[T\mathbb{E}_{q(U)}UU^\top\right] - 2}{\nu_{f,q}^{W^{-1}} + n + m - 2}\overline{K_{ff\cdot u}} + K_{fu}T\mathbb{V}_{q(U)}[U]TK_{uf} \tag{6.10}$$

$$= \frac{\nu_{f,q}^{W^{-1}} + n + \mathrm{Tr}\left[T\left(S_u + m_u m_u^\top\right)\right] - 2}{\nu_{f,q}^{W^{-1}} + n + m - 2}\overline{K_{ff\cdot u}} + K_{fu}TS_UTK_{uf}\,. \tag{6.11}$$

Therefore, under this approximate moments formulation during training we only need to sample from T to obtain gradients in the data fit term of the ELBO. Further work should explore this approach to obtain more stable gradients during optimization.

6.2.4 Bayesian Optimization & Active Learning

Throughout this thesis we have only explored the use of distributional uncertainty as a proxy for OOD detection. However, this is not the only use case.

Bayesian optimization (BO) represents a sequential design strategy for global optimization of black-box functions without assuming a functional form, being employed to optimize objective functions which are either computationally expensive or difficult to evaluate. Intuitively, one constructs a simpler surrogate model of the objective function, which will be used to solve the optimization problem. Some examples of the objective function can be quality control metrics on production lines (Maurya 2016), optimizing machine learning model hyperparameters to increase predictive capabilities (Turner et al. 2021), automatic chemical design (Gómez-Bombarelli et al. 2018) or pharmaceutical product development (Sano et al. 2020). Besides devising a surrogate model $\tilde{f}$ of the real objective function f, BO also implies using an acquisition function to guide in selecting future evaluation points. Common acquisition functions such Expected improvement or Upper Confidence Bound involves performing a tradeoff between exploitation and exploration. The latter involves evaluating the next point in unexplored regions flagged with high uncertainty. Hence, accurately capturing distributional uncertainty is vital for BO. GP (Shahriari et al. 2015) or sparse GP (McIntire et al. 2016) are the method of choice for the surrogate model due to their distance-awareness *properties. However, DGP might be better suited for modeling non-stationary objective functions compared to GP surrogates (Dutordoir et al. 2017; Hebbal et al. 2021). As we have seen in Chapter 3, DGP might fail in correctly propagating distributional uncertainty forward, especially in the case of high-dimensional data that gets projected down to a low dimensional latent space. DDGP suffer considerably less from this problem, hence they might constitute a better model choice for high-dimensional BO (Binois and Wycoff 2022).*

Furthermore, an arduous challenge in applied machine learning is in obtaining labelled data, which for some application domains can be quite costly, thus rendering the deployment of machine learning models unfeasible. Active Learning (Cohn et al. 1996) is the statistical framework by which a system learns from small datasets alongside providing users hints at which type of data should be

labelled to as to accelerate training and increase predictive capabilities. Pre-DL era ML models have been consistently used in this field (Settles 2009), while there has been a renewed interest in the machine learning community for active learning using BDL (Gal et al. 2017; Kirsch et al. 2019). GP and DGP have been used in the context of active learning (Fei et al. 2018; Kapoor et al. 2007). As previously mentioned, DGP might not constitute a reliable method for propagating distribution uncertainty forward in the hierarchy, hence DDGP can prove to be a viable solution for active learning due to its capability to learn from small amounts of data and to reliably estimate distributional uncertainty, which in this context will be used to decide which objects get labelled by an external oracle. In Chapter 4 we have dealt with biomedical image segmentation. Manually annotating medical images is a laborious task, hence active learning can be applied in this domain to efficiently annotate medical scans that bring the largest amount of information (Nath et al. 2020; Rahimi et al. 2021).

6.2.5 OOD detection & robustness to "in-the-wild" domain shifts

In Chapter 4 and associated supplementary material we have dealt with clear-cut outliers (e.g. tumors and white matter hyperintensities on brain scans; diverse set of vision diseases on retina scans) and have examined how well our proposed model is in detecting the pathological tissue. A drawback of our work is in the lack of assessing how robust our models are to different domains, such as in quantifying how accurate our segmentation pipeline is on healthy tissue nearby an anomalous tissue. This procedure can be applied if we also take into consideration failure prediction, which is the field of research that based on a given proxy (e.g., distributional uncertainty) can choose to reject predictions and defer them to an external oracle (e.g., clinical doctors in our biomedical segmentation scenario for further inspection). Furthermore, another limitation of our work is that our set of experiments correspond to domain generalization, where the training and test distributions comprise data from related but distinct domains. This does not encompass the entire spectrum of domain shifts. Additionally we also have subpopulation shift, where the testing distribution is a subpopulation of the training distribution. This type of domain shift is important to AI fairness as there is a well-documented literature on machine learning models performing abysmally on under-represented demographics (Koenecke et al. 2020). Future work should quantify the reliability of failure prediction and robustness to these type of shifts on real-life datasets (Koh et al. 2021; Malinin et al. 2021).

6.2.6 Conclusion

Gaussian Processes provide many advantages that recent machine learning models can benefit from, especially from the point of view of accurate uncertainty quantification. The question of whether deep Gaussian Processes can be used for OOD detection has been largely ignored till this current work. I hope that this thesis has provided the necessary insights and recipes for correctly using variants of DGP for detecting outliers, be it for tabular or image data. One obstacle that has prevented the large-scale adoption of Gaussian Processes (or in hybrid form, in conjunction with deep learning

architectures) has been the computational burden stemming from matrix inversion and log determinants. While this thesis has not solved this problem, I hope it took a step in the right direction of creating fast inverse-free Gaussian Processes.

References

Adam, V., Chang, P., Khan, M. E. E., and Solin, A. (2021). "Dual parameterization of sparse variational Gaussian processes". In: *Advances in Neural Information Processing Systems* 34.

Agueh, M. and Carlier, G. (2011). "Barycenters in the Wasserstein space". In: *SIAM Journal on Mathematical Analysis* 43.2, pp. 904–924.

Aitchison, L., Yang, A. X., and Ober, S. W. (2020). "Deep kernel processes". In: *arXiv preprint arXiv:2010.01590*.

Alfaro-Almagro, F. et al. (2018). "Image processing and Quality Control for the first 10,000 brain imaging datasets from UK Biobank". In: *Neuroimage* 166, pp. 400–424.

Alghadyan, A. A. (2011). "Diabetic retinopathy–An update". In: *Saudi Journal of Ophthalmology* 25.2, pp. 99–111.

Almgren, F. J. (1966). *Plateau's problem: an invitation to varifold geometry*. Vol. 13. American Mathematical Soc.

Álvarez-Esteban, P. C., Del Barrio, E, Cuesta-Albertos, J., and Matrán, C (2016). "A fixed-point approach to barycenters in Wasserstein space". In: *Journal of Mathematical Analysis and Applications* 441.2, pp. 744–762.

Amersfoort, J. van, Smith, L., Jesson, A., Key, O., and Gal, Y. (2021). "On feature collapse and deep kernel learning for single forward pass uncertainty". In: *arXiv preprint arXiv:2102.11409*.

Amersfoort, J. van, Smith, L., Teh, Y. W., and Gal, Y. (2020). "Simple and Scalable Epistemic Uncertainty Estimation Using a Single Deep Deterministic Neural Network". In: *arXiv preprint arXiv:2003.02037*.

Amini, A., Schwarting, W., Soleimany, A., and Rus, D. (2019). "Deep evidential regression". In: *arXiv preprint arXiv:1910.02600*.

Antonsanti, P.-L., Glaunès, J., Benseghir, T., Jugnon, V., and Kaltenmark, I. (2021). "Partial Matching in the Space of Varifolds". In: *International Conference on Information Processing in Medical Imaging*. Springer, pp. 123–135.

Arellano-Valle, R. B. and Genton, M. G. (2010). "Multivariate extended skew-t distributions and related families". In: *Metron* 68.3, pp. 201–234.

Azzalini, A. (2013). *The skew-normal and related families*. Vol. 3. Cambridge University Press.

Bachoc, F., Béthune, L., Gonzalez-Sanz, A., and Loubes, J.-M. (2023). "Gaussian Processes on Distributions based on Regularized Optimal Transport". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 4986–5010.

Bachoc, F., Gamboa, F., Loubes, J.-M., and Venet, N. (2017). "A Gaussian process regression model for distribution inputs". In: *IEEE Transactions on Information Theory* 64.10, pp. 6620–6637.

Bachoc, F., Suvorikova, A., Ginsbourger, D., Loubes, J.-M., and Spokoiny, V. (2018). "Gaussian processes with multidimensional distribution inputs via optimal transport and Hilbertian embedding". In: *arXiv preprint arXiv:1805.00753*.

Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J. S., Freymann, J. B., Farahani, K., and Davatzikos, C. (2017). "Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features". In: *Scientific data* 4, p. 170117.

Bakshi, R., Ariyaratana, S., Benedict, R. H., and Jacobs, L. (2001). "Fluid-attenuated inversion recovery magnetic resonance imaging detects cortical and juxtacortical multiple sclerosis lesions". In: *Archives of neurology* 58.5, pp. 742–748.

Bauer, M., Wilk, M. van der, and Rasmussen, C. E. (2016). "Understanding probabilistic sparse Gaussian process approximations". In: *Advances in neural information processing systems*, pp. 1533–1541.

Baumgartner, C. F., Tezcan, K. C., Chaitanya, K., Hötker, A. M., Muehlematter, U. J., Schawkat, K., Becker, A. S., Donati, O., and Konukoglu, E. (2019). "Phiseg: Capturing uncertainty in medical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 119–127.

Baur, C., Denner, S., Wiestler, B., Navab, N., and Albarqouni, S. (2021). "Autoencoders for unsupervised anomaly segmentation in brain MR images: a comparative study". In: *Medical Image Analysis*, p. 101952.

Baur, C., Wiestler, B., Albarqouni, S., and Navab, N. (2018). "Deep autoencoding models for unsupervised anomaly segmentation in brain MR images". In: *International MICCAI Brainlesion Workshop*. Springer, pp. 161–169.

Benavoli, A., Azzimonti, D., and Piga, D. (2020). "Skew Gaussian processes for classification". In: *Machine Learning* 109.9, pp. 1877–1902.

Binois, M. and Wycoff, N. (2022). "A survey on high-dimensional Gaussian process modeling with application to Bayesian optimization". In: *ACM Transactions on Evolutionary Learning and Optimization* 2.2, pp. 1–26.

Blomqvist, K., Kaski, S., and Heinonen, M. (2018). "Deep convolutional Gaussian processes". In: *arXiv preprint arXiv:1810.03052*.

Bodnar, T. and Okhrin, Y. (2008). "Properties of the singular, inverse and generalized inverse partitioned Wishart distributions". In: *Journal of Multivariate Analysis* 99.10, pp. 2389–2405.

Bosshard, T., Carambia, M, Goergen, K, Kotlarski, S, Krahe, P, Zappa, M, and Schär, C (2013). "Quantifying uncertainty sources in an ensemble of hydrological climate-impact projections". In: *Water Resources Research* 49.3, pp. 1523–1536.

Bradshaw, J., Matthews, A. G. d. G., and Ghahramani, Z. (2017). "Adversarial examples, uncertainty, and transfer testing robustness in Gaussian process hybrid deep networks". In: *arXiv preprint arXiv:1707.02476*.

Bruinsma, W., Perim, E., Tebbutt, W., Hosking, S., Solin, A., and Turner, R. (2020). "Scalable exact inference in multi-output gaussian processes". In: *International Conference on Machine Learning*. PMLR, pp. 1190–1201.

Bui, T., Hernández-Lobato, D., Hernandez-Lobato, J., Li, Y., and Turner, R. (2016). "Deep gaussian processes for regression using approximate expectation propagation". In: *International Conference on Machine Learning*, pp. 1472–1481.

Bulatov, Y. (2011). "Notmnist dataset". In: *Google (Books/OCR), Tech. Rep.[Online]. Available: http://yaroslavvb. blogspot. it/2011/09/notmnist-dataset. html* 2.

Castro, D. C., Tan, J., Kainz, B., Konukoglu, E., and Glocker, B. (2018). "Morpho-MNIST: Quantitative Assessment and Diagnostics for Representation Learning". In: *arXiv preprint arXiv:1809.10780*.

Charon, N., Charlier, B., Glaunès, J., Gori, P., and Roussillon, P. (2020). "Fidelity metrics between curves and surfaces: currents, varifolds, and normal cycles". In: *Riemannian geometric statistics in medical image analysis*. Elsevier, pp. 441–477.

Charon, N. and Trouvé, A. (2013). "The varifold representation of nonoriented shapes for diffeomorphic registration". In: *SIAM journal on Imaging Sciences* 6.4, pp. 2547–2580.

Charpentier, B., Borchert, O., Zügner, D., Geisler, S., and Günnemann, S. (2021). "Natural Posterior Network: Deep Bayesian Predictive Uncertainty for Exponential Family Distributions". In: *arXiv preprint arXiv:2105.04471*.

Charpentier, B., Zügner, D., and Günnemann, S. (2020). "Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts". In: *arXiv preprint arXiv:2006.09239*.

Chen, T., Fox, E., and Guestrin, C. (2014a). "Stochastic gradient hamiltonian monte carlo". In: *International conference on machine learning*. PMLR, pp. 1683–1691.

Chen, T., Dai, B., Wang, R., and Liu, D. (2014b). "Gaussian-process-based real-time ground segmentation for autonomous land vehicles". In: *Journal of Intelligent & Robotic Systems* 76.3, pp. 563–582.

Chen, X., Pawlowski, N., Glocker, B., and Konukoglu, E. (2019). "Unsupervised Lesion Detection with Locally Gaussian Approximation". In: *International Workshop on Machine Learning in Medical Imaging*. Springer, pp. 355–363.

– (2021). "Normative ascent with local gaussians for unsupervised lesion detection". In: *Medical Image Analysis*, p. 102208.

Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996). "Active learning with statistical models". In: *Journal of artificial intelligence research* 4, pp. 129–145.

Curth, A., Thoral, P., Wildenberg, W. van den, Bijlstra, P., Bruin, D. de, Elbers, P. W., and Fornasa, M. (2019). "Transferring Clinical Prediction Models Across Hospitals and Electronic Health Record Systems." In: *PKDD/ECML Workshops (1)*, pp. 605–621.

Cutajar, K., Bonilla, E. V., Michiardi, P., and Filippone, M. (2017). "Random feature expansions for deep Gaussian processes". In: *International Conference on Machine Learning*. PMLR, pp. 884–893.

Czolbe, S., Arnavaz, K., Krause, O., and Feragen, A. (2021). "Is segmentation uncertainty useful?" In: *International Conference on Information Processing in Medical Imaging*. Springer, pp. 715–726.

Damianou, A. (2015). "Deep Gaussian processes and variational propagation of uncertainty". PhD thesis. University of Sheffield.

Damianou, A. and Lawrence, N. (2013). "Deep gaussian processes". In: *Artificial Intelligence and Statistics*, pp. 207–215.

Damianou, A., Titsias, M., and Lawrence, N. (2011). "Variational Gaussian process dynamical systems". In: *Advances in neural information processing systems* 24.

Daston, L. and Galison, P. (1992). "The image of objectivity". In: *Representations* 40, pp. 81–128.

Davies, A. J. (2015). "Effective implementation of Gaussian process regression for machine learning". PhD thesis. University of Cambridge.

Dawid, A. P. (1981). "Some matrix-variate distribution theory: notational considerations and a Bayesian application". In: *Biometrika* 68.1, pp. 265–274.

Depeweg, S., Hernandez-Lobato, J.-M., Doshi-Velez, F., and Udluft, S. (2018). "Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning". In: *International Conference on Machine Learning*. PMLR, pp. 1184–1193.

Der Kiureghian, A. and Ditlevsen, O. (2009). "Aleatory or epistemic? Does it matter?" In: *Structural safety* 31.2, pp. 105–112.

Dey, R. and Hong, Y. (2021). "ASC-Net: Adversarial-Based Selective Network for Unsupervised Anomaly Segmentation". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*. Ed. by Bruijne, M. de, Cattin, P. C., Cotin, S., Padoy, N., Speidel, S., Zheng, Y., and Essert, C. Cham: Springer International Publishing, pp. 236–247. ISBN: 978-3-030-87240-3.

Ding, N., Qi, Y., and Vishwanathan, S. (2011). "t-divergence based approximate inference". In: *Advances in Neural Information Processing Systems* 24.

Ding, P. (2016). "On the conditional distribution of the multivariate t distribution". In: *The American Statistician* 70.3, pp. 293–295.

Domingues, R., Michiardi, P., Zouaoui, J., and Filippone, M. (2018). "Deep gaussian process autoencoders for novelty detection". In: *Machine Learning* 107.8, pp. 1363–1383.

Dowson, D. and Landau, B. (1982). "The Fréchet distance between multivariate normal distributions". In: *Journal of Multivariate Analysis* 12, pp. 450–455.

Dunlop, M. M., Girolami, M. A., Stuart, A. M., and Teckentrup, A. L. (2018). "How deep are deep Gaussian processes?" In: *Journal of Machine Learning Research* 19.54, pp. 1–46.

Dutordoir, V., Knudde, N., Herten, J. van der, Couckuyt, I., and Dhaene, T. (2017). "Deep Gaussian process metamodeling of sequentially sampled non-stationary response surfaces". In: *2017 Winter Simulation Conference (WSC)*. IEEE, pp. 1728–1739.

Dutordoir, V., Wilk, M. van der, Artemev, A., and Hensman, J. (2020). "Bayesian Image Classification with Deep Convolutional Gaussian Processes". In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Ed. by Chiappa, S. and Calandra, R. Vol. 108. Proceedings of Machine Learning Research. PMLR, pp. 1529–1539.

Dutordoir, V., Wilk, M. van der, Artemev, A., Tomczak, M., and Hensman, J. (2019). "Translation Insensitivity for Deep Convolutional Gaussian Processes". In: *arXiv preprint arXiv:1902.05888*.

Duvenaud, D., Rippel, O., Adams, R., and Ghahramani, Z. (2014). "Avoiding pathologies in very deep networks". In: *Artificial Intelligence and Statistics*. PMLR, pp. 202–210.

Empiricus, S. (1933). "Outlines of pyrrhonism". In: *Eng. trans, by RG Bury (Cambridge Mass.: Harvard UP, 1976), II* 20, pp. 90–1.

Evans, T. and Nair, P. (2018). "Scalable Gaussian processes with grid-structured eigenfunctions (GP-GRIEF)". In: *International Conference on Machine Learning*. PMLR, pp. 1417–1426.

Fei, J., Zhao, J., Sun, S., and Liu, Y. (2018). "Active Learning Methods with Deep Gaussian Processes". In: *Neural Information Processing*. Ed. by Cheng, L., Leung, A. C. S., and Ozawa, S. Cham: Springer International Publishing, pp. 473–483. ISBN: 978-3-030-04182-3.

Filippone, M. and Engler, R. (2015b). "Enabling scalable stochastic gradient-based inference for Gaussian processes by employing the Unbiased LInear System SolvEr (ULISSE)". In: *International Conference on Machine Learning*. PMLR, pp. 1015–1024.

– (2015a). "Enabling scalable stochastic gradient-based inference for Gaussian processes by employing the Unbiased LInear System SolvEr (ULISSE)". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Bach, F. and Blei, D. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 1015–1024.

Foong, A. Y., Li, Y., Hernández-Lobato, J. M., and Turner, R. E. (2019). "'In-Between'Uncertainty in Bayesian Neural Networks". In: *arXiv preprint arXiv:1906.11537*.

Fox, C. R. and Ülkümen, G. (2011). "Distinguishing two dimensions of uncertainty". In: *Fox, Craig R. and Gülden Ülkümen (2011),"Distinguishing Two Dimensions of Uncertainty," in Essays in Judgment and Decision Making, Brun, W., Kirkebøen, G. and Montgomery, H., eds. Oslo: Universitetsforlaget*.

Franchi, G., Bursuc, A., Aldea, E., Dubuisson, S., and Bloch, I. (2020). "One Versus all for deep Neural Network Incertitude (OVNNI) quantification". In: *arXiv preprint arXiv:2006.00954*.

Futami, F., Sato, I., and Sugiyama, M. (2017). "Expectation propagation for t-exponential family using q-algebra". In: *Advances in Neural Information Processing Systems* 30.

Gal, Y. and Ghahramani, Z. (2016a). "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". In: *international conference on machine learning*. PMLR, pp. 1050–1059.

– (2016b). "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *ICML*.

Gal, Y., Islam, R., and Ghahramani, Z. (2017). "Deep bayesian active learning with image data". In: *International Conference on Machine Learning*. PMLR, pp. 1183–1192.

Gallier, J. et al. (2010). "The Schur complement and symmetric positive semidefinite (and definite) matrices". In: *Penn Engineering*, pp. 1–12.

Garaud, D. and Mallet, V. (2011). "Automatic calibration of an ensemble for uncertainty estimation and probabilistic forecast: Application to air quality". In: *Journal of Geophysical Research: Atmospheres* 116.D19.

Gardner, J., Pleiss, G., Wu, R., Weinberger, K., and Wilson, A. (2018a). "Product kernel interpolation for scalable Gaussian processes". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1407–1416.

Gardner, J. R., Pleiss, G., Bindel, D., Weinberger, K. Q., and Wilson, A. G. (2018b). "Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration". In: *arXiv preprint arXiv:1809.11165*.

Garriga-Alonso, A., Aitchison, L., and Rasmussen, C. E. (2018). "Deep convolutional networks as shallow gaussian processes". In: *arXiv preprint arXiv:1808.05587*.

Girard, A, Rasmussen, C., and Murray-Smith, R (2002). *Gaussian process priors with uncertain inputs: Multiple-step ahead prediction. Univ. Glasgow*. Tech. rep. Glasgow, Technical Report TR-2002-119.

Girard, A. (2004). *Approximate methods for propagation of uncertainty with Gaussian process models*. University of Glasgow (United Kingdom).

Girard, D. (1987). *Un algorithme simple et rapide pour la validation croisée généralisée sur des problèmes de grande taille*.

Gómez-Bombarelli, R. et al. (2018). "Automatic chemical design using a data-driven continuous representation of molecules". In: *ACS central science* 4.2, pp. 268–276.

Granström, K. and Orguner, U. (2012). "On the reduction of Gaussian inverse Wishart mixtures". In: *2012 15th International Conference on Information Fusion*. IEEE, pp. 2162–2169.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). "A kernel two-sample test". In: *The Journal of Machine Learning Research* 13.1, pp. 723–773.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). "On calibration of modern neural networks". In: *arXiv preprint arXiv:1706.04599*.

Ha, D., Dai, A., and Le, Q. V. (2016). "Hypernetworks". In: *arXiv preprint arXiv:1609.09106*.

Hafner, D., Tran, D., Lillicrap, T., Irpan, A., and Davidson, J. (2020). "Noise contrastive priors for functional uncertainty". In: *Uncertainty in Artificial Intelligence*. PMLR, pp. 905–914.

Havasi, M., Lobato, J. M. H., and Fuentes, J. J. M. (2018). "Inference in Deep Gaussian Processes using Stochastic Gradient Hamiltonian Monte Carlo". In: *arXiv preprint arXiv:1806.05490*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Identity mappings in deep residual networks". In: *European conference on computer vision*. Springer, pp. 630–645.

Hebbal, A., Brevault, L., Balesdent, M., Talbi, E.-G., and Melab, N. (2021). "Bayesian optimization using deep Gaussian processes with applications to aerospace system design". In: *Optimization and Engineering* 22.1, pp. 321–361.

Hendrycks, D. and Gimpel, K. (2016). "A baseline for detecting misclassified and out-of-distribution examples in neural networks". In: *arXiv preprint arXiv:1610.02136*.

Henning, C., D'Angelo, F., and Grewe, B. F. (2021). "Are Bayesian neural networks intrinsically good at out-of-distribution detection?" In: *arXiv preprint arXiv:2107.12248*.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). "Gaussian processes for big data". In: *arXiv preprint arXiv:1309.6835*.

Hensman, J., Matthews, A., and Ghahramani, Z. (2015). "Scalable variational Gaussian process classification". In: *Artificial Intelligence and Statistics*. PMLR, pp. 351–360.

Hernández-Lobato, D. and Hernández-Lobato, J. M. (2016). "Scalable Gaussian process classification via expectation propagation". In: *Artificial Intelligence and Statistics*. PMLR, pp. 168–176.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). "Stochastic variational inference". In: *The Journal of Machine Learning Research* 14.1, pp. 1303–1347.

Hoover, A., Kouznetsova, V., and Goldbaum, M. (2000). "Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response". In: *IEEE Transactions on Medical imaging* 19.3, pp. 203–210.

Howell, W. C. and Burnett, S. A. (1978). "Uncertainty measurement: A cognitive taxonomy". In: *Organizational Behavior and Human Performance* 22.1, pp. 45–68.

Hu, S., Worrall, D., Knegt, S., Veeling, B., Huisman, H., and Welling, M. (2019). "Supervised uncertainty quantification for segmentation with multiple annotations". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 137–145.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708.

Hüllermeier, E. and Waegeman, W. (2021). "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods". In: *Machine Learning* 110, pp. 457–506.

Huszár, F. (2017). "Variational inference using implicit distributions". In: *arXiv preprint arXiv:1702.08235*.

Hutchinson, M. F. (1989). "A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines". In: *Communications in Statistics-Simulation and Computation* 18.3, pp. 1059–1076.

Imai, S., Takekuma, Y., Kashiwagi, H., Miyai, T., Kobayashi, M., Iseki, K., and Sugawara, M. (2020). "Validation of the usefulness of artificial neural networks for risk prediction of adverse drug reactions used for individual patients in clinical practice". In: *Plos one* 15.7, e0236789.

Jacobsen, J.-H., Behrmann, J., Zemel, R., and Bethge, M. (2018). "Excessive invariance causes adversarial vulnerability". In: *arXiv preprint arXiv:1811.00401*.

Jankowiak, M., Pleiss, G., and Gardner, J. (2020a). "Deep sigma point processes". In: *Conference on Uncertainty in Artificial Intelligence*. PMLR, pp. 789–798.

– (2020b). "Parametric Gaussian process regressors". In: *International Conference on Machine Learning*. PMLR, pp. 4702–4712.

Jebara, T., Kondor, R., and Howard, A. (2004). "Probability product kernels". In: *The Journal of Machine Learning Research* 5, pp. 819–844.

Jiang, X., Osl, M., Kim, J., and Ohno-Machado, L. (2012). "Calibrating predictive model estimates to support personalized medicine". In: *Journal of the American Medical Informatics Association* 19.2, pp. 263–274.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). "An introduction to variational methods for graphical models". In: *Machine learning* 37.2, pp. 183–233.

Jylänki, P., Vanhatalo, J., and Vehtari, A. (2011). "Robust Gaussian Process Regression with a Student-t Likelihood." In: *Journal of Machine Learning Research* 12.11.

Kahneman, D. and Tversky, A. (1982). "Variants of uncertainty". In: *Cognition* 11.2, pp. 143–157.

Kaltenmark, I., Charlier, B., and Charon, N. (2017). "A general framework for curve and surface comparison and registration with oriented varifolds". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3346–3355.

Kapoor, A., Grauman, K., Urtasun, R., and Darrell, T. (2007). "Active learning with gaussian processes for object categorization". In: *2007 IEEE 11th International Conference on Computer Vision*. IEEE, pp. 1–8.

Kendall, A., Badrinarayanan, V., and Cipolla, R. (2015). "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding". In: *arXiv preprint arXiv:1511.02680*.

Kendall, A. and Gal, Y. (2017). "What uncertainties do we need in bayesian deep learning for computer vision?" In: *arXiv preprint arXiv:1703.04977*.

Khan, E., Mohamed, S., and Murphy, K. P. (2012). "Fast Bayesian inference for non-conjugate Gaussian process regression". In: *Advances in Neural Information Processing Systems* 25.

Khan, M. and Lin, W. (2017). "Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models". In: *Artificial Intelligence and Statistics*. PMLR, pp. 878–887.

King, N. J. and Lawrence, N. D. (2006). "Fast variational inference for Gaussian process models through KL-correction". In: *European Conference on Machine Learning*. Springer, pp. 270–281.

Kingma, D. P., Salimans, T., and Welling, M. (2015). "Variational dropout and the local reparameterization trick". In: *Advances in neural information processing systems*, pp. 2575–2583.

Kirsch, A., Van Amersfoort, J., and Gal, Y. (2019). "Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning". In: *Advances in neural information processing systems* 32, pp. 7026–7037.

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

Koenecke, A. et al. (2020). "Racial disparities in automated speech recognition". In: *Proceedings of the National Academy of Sciences* 117.14, pp. 7684–7689.

Koh, P. W. et al. (2021). "Wilds: A benchmark of in-the-wild distribution shifts". In: *International Conference on Machine Learning*. PMLR, pp. 5637–5664.

Kohl, S., Romera-Paredes, B., Meyer, C., De Fauw, J., Ledsam, J. R., Maier-Hein, K., Eslami, S. A., Rezende, D. J., and Ronneberger, O. (2018). "A probabilistic u-net for segmentation of ambiguous images". In: *Advances in Neural Information Processing Systems*, pp. 6965–6975.

Kolouri, S., Park, S. R., Thorpe, M., Slepcev, D., and Rohde, G. K. (2017). "Optimal mass transport: Signal processing and machine-learning applications". In: *IEEE signal processing magazine* 34.4, pp. 43–59.

Kolouri, S., Zou, Y., and Rohde, G. K. (2016). "Sliced Wasserstein kernels for probability distributions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5258–5267.

Kotz, S. and Nadarajah, S. (2004). *Multivariate t-distributions and their applications*. Cambridge University Press.

Krizhevsky, A. (2009). *Learning multiple layers of features from tiny images*. Tech. rep.

Kumar, V., Singh, V., Srijith, P., and Damianou, A. (2018). "Deep gaussian processes with convolutional kernels". In: *arXiv preprint arXiv:1806.01655*.

Kurakin, A., Goodfellow, I. J., and Bengio, S. (2018). "Adversarial examples in the physical world". In: *Artificial intelligence safety and security*. Chapman and Hall/CRC, pp. 99–112.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *Advances in neural information processing systems* 30.

Lalchand, V. and Rasmussen, C. E. (2020). "Approximate inference for fully Bayesian Gaussian process regression". In: *Symposium on Advances in Approximate Bayesian Inference*. PMLR, pp. 1–12.

Lawrence, N. (2003). "Gaussian process latent variable models for visualisation of high dimensional data". In: *Advances in neural information processing systems* 16.

Lawrence, N. D. and Moore, A. J. (2007). "Hierarchical Gaussian process latent variable models". In: *Proceedings of the 24th international conference on Machine learning*, pp. 481–488.

LeCun, Y. (1998). "The MNIST database of handwritten digits". In: *http://yann. lecun. com/exdb/mnist/*.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE*, pp. 2278–2324.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. (2017). "Deep neural networks as gaussian processes". In: *arXiv preprint arXiv:1711.00165*.

Liang, S., Li, Y., and Srikant, R. (2017). "Enhancing the reliability of out-of-distribution image detection in neural networks". In: *arXiv preprint arXiv:1706.02690*.

Lindinger, J., Reeb, D., Lippert, C., and Rakitsch, B. (2020). "Beyond the mean-field: Structured deep Gaussian processes improve the predictive uncertainties". In: *Advances in Neural Information Processing Systems* 33, pp. 8498–8509.

Liu, J., Paisley, J., Kioumourtzoglou, M.-A., and Coull, B. (2019). "Accurate uncertainty estimation and decomposition in ensemble learning". In: *Advances in Neural Information Processing Systems*, pp. 8952–8963.

Liu, J. Z., Lin, Z., Padhy, S., Tran, D., Bedrax-Weiss, T., and Lakshminarayanan, B. (2020a). "Simple and principled uncertainty estimation with deterministic deep learning via distance awareness". In: *arXiv preprint arXiv:2006.10108*.

Liu, S., Sun, X., Ramadge, P. J., and Adams, R. P. (2020b). "Task-agnostic amortized inference of Gaussian Process hyperparameters". In: *Advances in Neural Information Processing Systems* 33, pp. 21440–21452.

Lu, C.-K. and Shafto, P. (2021). "Conditional Deep Gaussian Processes: Empirical Bayes Hyperdata Learning". In: *Entropy* 23.11, p. 1387.

Malinin, A. and Gales, M. (2018). "Predictive uncertainty estimation via prior networks". In: *Advances in Neural Information Processing Systems*, pp. 7047–7058.

Malinin, A. et al. (2021). "Shifts: A dataset of real distributional shift across multiple large-scale tasks". In: *arXiv preprint arXiv:2107.07455*.

Mallasto, A. and Feragen, A. (2017). "Learning from uncertain curves: The 2-Wasserstein metric for Gaussian processes". In: *Advances in Neural Information Processing Systems* 30.

Maroñas, J., Hamelijnck, O., Knoblauch, J., and Damoulas, T. (2021). "Transforming Gaussian processes with normalizing flows". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1081–1089.

Mårtensson, G. et al. (2020). "The reliability of a deep learning model in clinical out-of-distribution MRI data: a multicohort study". In: *Medical Image Analysis* 66, p. 101714.

Maurya, A. (2016). "Bayesian optimization for predicting rare internal failures in manufacturing processes". In: *2016 IEEE international conference on big data (big data)*. IEEE, pp. 2036–2045.

McHutchon, A. and Rasmussen, C. (2011). "Gaussian process training with input noise". In: *Advances in neural information processing systems* 24.

McIntire, M., Ratner, D., and Ermon, S. (2016). "Sparse Gaussian Processes for Bayesian Optimization." In: *UAI*.

Meyer-Eppler, W. (1958). "Statistic and psychologic problems of sound". In: *Die Reihe* 1, pp. 55–61.

Milios, D., Camoriano, R., Michiardi, P., Rosasco, L., and Filippone, M. (2018). "Dirichlet-based gaussian processes for large-scale calibrated classification". In: *arXiv preprint arXiv:1805.10915*.

Minka, T. P. (2013). "Expectation propagation for approximate Bayesian inference". In: *arXiv preprint arXiv:1301.2294*.

Modeste, T. and Dombry, C. (2022). "Characterization of translation invariant MMD on R d and connections with Wasserstein distances". In.

Molchanov, D., Kharitonov, V., Sobolev, A., and Vetrov, D. (2019). "Doubly semi-implicit variational inference". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 2593–2602.

Molchanova, I., Molchanov, D., Quadrianto, N., and Vetrov, D. (2019). "Structured Semi-Implicit Variational Inference". In.

Monteiro, M., Folgoc, L. L., Castro, D. C. de, Pawlowski, N., Marques, B., Kamnitsas, K., Wilk, M. van der, and Glocker, B. (2020). "Stochastic Segmentation Networks: Modelling Spatially Correlated Aleatoric Uncertainty". In: *arXiv preprint arXiv:2006.06015*.

Morales-Alvarez, P., Hernández-Lobato, D., Molina, R., and Hernández-Lobato, J. M. (2020). "Activation-level uncertainty in deep neural networks". In: *International Conference on Learning Representations*.

Muirhead, R. J. (1982). "Aspects of multivariate statistical analysis." In: *JOHN WILEY & SONS, INC., 605 THIRD AVE., NEW YORK, NY 10158, USA, 1982, 656*.

Nair, T., Precup, D., Arnold, D. L., and Arbel, T. (2020). "Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation". In: *Medical image analysis* 59, p. 101557.

Nath, V., Yang, D., Landman, B. A., Xu, D., and Roth, H. R. (2020). "Diminishing Uncertainty within the Training Pool: Active Learning for Medical Image Segmentation". In: *IEEE Transactions on Medical Imaging*.

Neal, R. M. (1997). "Monte Carlo implementation of Gaussian process models for Bayesian regression and classification". In: *arXiv preprint physics/9701026*.

– (2012). *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). "Reading digits in natural images with unsupervised feature learning". In.

Nguyen, T. V., Bonilla, E. V., et al. (2014). "Collaborative Multi-output Gaussian Processes." In: *UAI*. Citeseer, pp. 643–652.

Ober, S. W. and Aitchison, L. (2021). "Global inducing point variational posteriors for bayesian neural networks and deep gaussian processes". In: *International Conference on Machine Learning*. PMLR, pp. 8248–8259.

Ober, S. W., Rasmussen, C. E., and Wilk, M. van der (2021). "The Promises and Pitfalls of Deep Kernel Learning". In: *arXiv preprint arXiv:2102.12108*.

Paananen, T., Piironen, J., Andersen, M. R., and Vehtari, A. (2019). "Variable selection for Gaussian processes via sensitivity analysis of the posterior predictive distribution". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1743–1752.

Padhy, S., Nado, Z., Ren, J., Liu, J., Snoek, J., and Lakshminarayanan, B. (2020). "Revisiting One-vs-All Classifiers for Predictive Uncertainty and Out-of-Distribution Detection in Neural Networks". In: *arXiv preprint arXiv:2007.05134*.

Pawlowski, N., Brock, A., Lee, M. C., Rajchl, M., and Glocker, B. (2017). "Implicit weight uncertainty in neural networks". In: *arXiv preprint arXiv:1711.01297*.

Pinaya, W. H. L., Tudosiu, P.-D., Gray, R., Rees, G., Nachev, P., Ourselin, S., and Cardoso, M. J. (2021). "Unsupervised Brain Anomaly Detection and Segmentation with Transformers". In: *Proceedings of the Fourth Conference on Medical Imaging with Deep Learning*. Ed. by Heinrich, M., Dou, Q., Bruijne, M. de, Lellmann, J., Schläfer, A., and Ernst, F. Vol. 143. Proceedings of Machine Learning Research. PMLR, pp. 596–617.

Popescu, S. G., Glocker, B., and Wilk, M. Van der (2022). "Matrix Inversion free variational inference in Conditional Student's T Processes". In: *Symposium on Advances in Approximate Bayesian Inference*.

Popescu, S. G., Sharp, D., Cole, J., and Glocker, B. (2020). "Hierarchical Gaussian Processes with Wasserstein-2 Kernels". In: *NeurIPS Workshop on Bayesian Deep Learning*.

Popescu, S. G., Sharp, D. J., Cole, J. H., Kamnitsas, K., and Glocker, B. (2021). "Distributional Gaussian Process Layers for Outlier Detection in Image Segmentation". In: *International Conference on Information Processing in Medical Imaging*. Springer, pp. 415–427.

Porwal, P., Pachade, S., Kamble, R., Kokare, M., Deshmukh, G., Sahasrabuddhe, V., and Meriaudeau, F. (2018). "Indian diabetic retinopathy image dataset (IDRiD): a database for diabetic retinopathy screening research". In: *Data* 3.3, p. 25.

Potapczynski, A., Wu, L., Biderman, D., Pleiss, G., and Cunningham, J. P. (2021). "Bias-Free Scalable Gaussian Processes via Randomized Truncations". In: *arXiv preprint arXiv:2102.06695*.

Pugachev, V. S. (2013). *Theory of Random Functions: And Its Application to Control Problems*. Vol. 5. Elsevier.

Quinonero-Candela, J., Girard, A., and Rasmussen, C. E. (2003). "Prediction at an uncertain input for Gaussian processes and relevance vector machines-application to multiple-step ahead time-series forecasting". In.

Quinonero-Candela, J. and Rasmussen, C. E. (2005). "A unifying view of sparse approximate Gaussian process regression". In: *The Journal of Machine Learning Research* 6, pp. 1939–1959.

Quinonero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. (2008). *Dataset shift in machine learning*. Mit Press.

Rahimi, A. and Recht, B. (2007). "Random features for large-scale kernel machines". In: *Advances in neural information processing systems* 20.

Rahimi, S., Oktay, O., Alvarez-Valle, J., and Bharadwaj, S. (2021). "Addressing the Exorbitant Cost of Labeling Medical Images with Active Learning". In: *International Conference on Machine Learning in Medical Imaging and Analysis*. Best Presentation Award.

Rezende, D. and Mohamed, S. (2015). "Variational inference with normalizing flows". In: *International conference on machine learning*. PMLR, pp. 1530–1538.

Robinson, R. D. (July 2020). "Reliable Machine Learning for Medical Imaging Data through Automated Quality Control and Data Harmonization". In.

Roetzer, G. R. (2020). "Efficient and scalable inference for generalized student-T process models". PhD thesis. Trinity College Dublin. School of Computer Science & Statistics. Discipline ...

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, pp. 234–241.

Rosca, M., Weber, T., Gretton, A., and Mohamed, S. (2020). "A case for new neural network smoothness constraints". In.

Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., and Müller, K.-R. (2021). "A unifying review of deep and shallow anomaly detection". In: *Proceedings of the IEEE*.

Salimbeni, H. and Deisenroth, M. (2017). "Doubly stochastic variational inference for deep gaussian processes". In: *Advances in Neural Information Processing Systems*, pp. 4588–4599.

Salimbeni, H., Eleftheriadis, S., and Hensman, J. (2018). "Natural gradients in practice: Non-conjugate variational inference in Gaussian process models". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 689–697.

Sano, S., Kadowaki, T., Tsuda, K., and Kimura, S. (2020). "Application of Bayesian optimization for pharmaceutical product development". In: *Journal of Pharmaceutical Innovation* 15.3, pp. 333–343.

Särkkä, S. (2013). *Bayesian filtering and smoothing*. 3. Cambridge University Press.

Schlegl, T., Seeböck, P., Waldstein, S. M., Langs, G., and Schmidt-Erfurth, U. (2019). "f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks". In: *Medical image analysis* 54, pp. 30–44.

Seeger, M., Williams, C., and Lawrence, N. (2003). *Fast forward selection to speed up sparse Gaussian process regression*. Tech. rep.

Sensoy, M., Kaplan, L., and Kandemir, M. (2018). "Evidential deep learning to quantify classification uncertainty". In: *arXiv preprint arXiv:1806.01768*.

Settles, B. (2009). "Active learning literature survey". In.

Shah, A., Wilson, A., and Ghahramani, Z. (2014). "Student-t processes as alternatives to Gaussian processes". In: *Artificial intelligence and statistics*, pp. 877–885.
Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. (2015). "Taking the human out of the loop: A review of Bayesian optimization". In: *Proceedings of the IEEE* 104.1, pp. 148–175.
Shang, L. and Chan, A. B. (2013). "On approximate inference for generalized Gaussian process models". In: *arXiv preprint arXiv:1311.6371*.
Sheth, R., Wang, Y., and Khardon, R. (2015). "Sparse variational inference for generalized GP models". In: *International Conference on Machine Learning*. PMLR, pp. 1302–1311.
Snelson, E. and Ghahramani, Z. (2005). "Sparse Gaussian processes using pseudo-inputs". In: *Advances in neural information processing systems* 18.
– (2006). "Sparse Gaussian processes using pseudo-inputs". In: *Advances in neural information processing systems*, pp. 1257–1264.
Sobolev, A. and Vetrov, D. (2019). "Importance weighted hierarchical variational inference". In: *arXiv preprint arXiv:1905.03290*.
Sokolić, J., Giryes, R., Sapiro, G., and Rodrigues, M. R. (2017). "Robust large margin deep neural networks". In: *IEEE Transactions on Signal Processing* 65.16, pp. 4265–4280.
Solin, A. and Särkkä, S. (2015). "State space methods for efficient inference in Student-t process regression". In: *Artificial Intelligence and Statistics*. PMLR, pp. 885–893.
Staal, J., Abràmoff, M. D., Niemeijer, M., Viergever, M. A., and Van Ginneken, B. (2004). "Ridge-based vessel segmentation in color images of the retina". In: *IEEE transactions on medical imaging* 23.4, pp. 501–509.
Stefan, R. C. and Cheche, T. O. (2016). "Coin toss modeling". In: *arXiv preprint arXiv:1612.06705*.
Strzałko, J., Grabski, J., Stefański, A, Perlikowski, P., and Kapitaniak, T. (2008). "Dynamics of coin tossing is predictable". In: *Physics reports* 469.2, pp. 59–92.
Sweeney, E., Shinohara, R., Shea, C., Reich, D., and Crainiceanu, C. M. (2013). "Automatic lesion incidence estimation and detection in multiple sclerosis using multisequence longitudinal MRI". In: *American Journal of Neuroradiology* 34.1, pp. 68–73.
Takatsu, A. (2011). "Wasserstein geometry of Gaussian measures". In.
Tang, X. (2019). "The role of artificial intelligence in medical imaging research". In: *BJR| Open* 2.1, p. 20190031.
Thi Thien Trang, B., Loubes, J.-M., Risser, L., and Balaresque, P. (2019). "Distribution regression model with a Reproducing Kernel Hilbert Space approach". In: *Communications in Statistics-Theory and Methods*, pp. 1–23.
Timonen, J., Mannerström, H., Vehtari, A., and Lähdesmäki, H. (2019). "lgpr: An interpretable non-parametric method for inferring covariate effects from longitudinal data". In: *arXiv preprint arXiv:1912.03549*.
Tipping, M. E. and Lawrence, N. D. (2005). "Variational inference for Student-t models: Robust Bayesian interpolation and generalised component analysis". In: *Neurocomputing* 69.1-3, pp. 123–141.
Titsias, M. (2009). "Variational learning of inducing variables in sparse Gaussian processes". In: *Artificial Intelligence and Statistics*, pp. 567–574.
Titsias, M. and Lawrence, N. D. (2010). "Bayesian Gaussian process latent variable model". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, pp. 844–851.
Torralba, A. and Efros, A. A. (2011). "Unbiased look at dataset bias". In: *CVPR 2011*. IEEE, pp. 1521–1528.
Tran, P. et al. (2022). "Automatic segmentation of white matter hyperintensities: validation and comparison with state-of-the-art methods on both Multiple Sclerosis and elderly subjects". In: *NeuroImage: Clinical* 33, p. 102940.

Tsallis, C. (1988). "Possible generalization of Boltzmann-Gibbs statistics". In: *Journal of statistical physics* 52, pp. 479–487.
Tsallis, C. and Brigatti, E. (2004). "Nonextensive statistical mechanics: A brief introduction". In: *Continuum Mechanics and Thermodynamics* 16.3, pp. 223–235.
Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., Xu, Z., and Guyon, I. (2021). "Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020". In: *arXiv preprint arXiv:2104.10201*.
Ülkümen, G., Fox, C. R., and Malle, B. F. (2016). "Two dimensions of subjective uncertainty: Clues from natural language." In: *Journal of experimental psychology: General* 145.10, p. 1280.
Ulmer, D. T., Hardmeier, C., and Frellsen, J. (2023). "Prior and posterior networks: A survey on evidential deep learning methods for uncertainty estimation". In: *Transactions on Machine Learning Research*.
Ustyuzhaninov, I., Kazlauskaite, I., Kaiser, M., Bodin, E., Campbell, N. D., and Ek, C. H. (2019). "Compositional uncertainty in deep Gaussian processes". In: *arXiv preprint arXiv:1909.07698*.
Vanhatalo, J., Jylänki, P., and Vehtari, A. (2009). "Gaussian process regression with Student-t likelihood". In: *Advances in neural information processing systems* 22, pp. 1910–1918.
Villani, C. (2008). *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media.
Wang, D. et al. (2020). "A statistical framework for assessing pharmacological responses and biomarkers using uncertainty estimates". In: *Elife* 9, e60352.
Wang, J., Hertzmann, A., and Fleet, D. J. (2005). "Gaussian process dynamical models". In: *Advances in neural information processing systems* 18.
Wang, K., Pleiss, G., Gardner, J., Tyree, S., Weinberger, K. Q., and Wilson, A. G. (2019). "Exact Gaussian processes on a million data points". In: *Advances in Neural Information Processing Systems* 32, pp. 14648–14659.
Weng, T.-W., Zhang, H., Chen, P.-Y., Yi, J., Su, D., Gao, Y., Hsieh, C.-J., and Daniel, L. (2018). "Evaluating the robustness of neural networks: An extreme value theory approach". In: *arXiv preprint arXiv:1801.10578*.
Wilk, M. van der, Dutordoir, V., John, S., Artemev, A., Adam, V., and Hensman, J. (2020a). "A framework for interdomain and multioutput Gaussian processes". In: *arXiv preprint arXiv:2003.01115*.
Wilk, M. van der, John, S., Artemev, A., and Hensman, J. (2020b). "Variational Gaussian Process Models without Matrix Inverses". In: *Proceedings of The 2nd Symposium on Advances in Approximate Bayesian Inference*. Ed. by Zhang, C., Ruiz, F., Bui, T., Dieng, A. B., and Liang, D. Vol. 118. Proceedings of Machine Learning Research. PMLR, pp. 1–9.
Wilk, M. Van der, Rasmussen, C. E., and Hensman, J. (2017). "Convolutional gaussian processes". In: *Advances in Neural Information Processing Systems*, pp. 2849–2858.
Williams, C. K. and Barber, D. (1998). "Bayesian classification with Gaussian processes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.12, pp. 1342–1351.
Wilson, A. and Nickisch, H. (2015). "Kernel interpolation for scalable structured Gaussian processes (KISS-GP)". In: *International Conference on Machine Learning*. PMLR, pp. 1775–1784.
Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. (2016). "Deep kernel learning". In: *Artificial Intelligence and Statistics*, pp. 370–378.
Wilson, A. G. and Izmailov, P. (2020). "Bayesian deep learning and a probabilistic perspective of generalization". In: *arXiv preprint arXiv:2002.08791*.
Yin, M. and Zhou, M. (2018). "Semi-implicit variational inference". In: *International Conference on Machine Learning*. PMLR, pp. 5660–5669.
Yip, S., Ferro, C. A., Stephenson, D. B., and Hawkins, E. (2011). "A simple, coherent framework for partitioning uncertainty in climate predictions". In: *Journal of climate* 24.17, pp. 4634–4643.

Yu, H., Chen, Y., Dai, Z., Low, K. H., and Jaillet, P. (2019). "Implicit Posterior Variational Inference for Deep Gaussian Processes". In: *arXiv preprint arXiv:1910.11998*.

Zhou, S. K., Greenspan, H., Davatzikos, C., Duncan, J. S., Van Ginneken, B., Madabhushi, A., Prince, J. L., Rueckert, D., and Summers, R. M. (2021). "A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises". In: *Proceedings of the IEEE*.

Appendices

A Further details pertaining to Distributional Deep Gaussian Processes

A.1 Varifold Kernels

Varifolds were first introduced in Geometric Measure Theory (GMT) (Almgren 1966), in which shapes like submanifolds are represented in certain spaces of generalized measures such as currents, varifolds or normal cycles. This general framework was adapted to suit requirements for computational anatomy matching in Charon and Trouvé (2013) and without loss of generality, throughout the remainder of this subsection we will focus solely on surfaces embedded in $\mathbb{R}^3$ and how to match them.

We commence by introducing elementary definitions from GMT. Subsequently introduced methods are capable of working with smooth surfaces alongside piecewise smooth or discrete objects. The umbrella term of rectifiable subsets in GMT encompasses the aforementioned scenarios. We denote $\mathbb{M} \in \mathbb{R}^3$ a rectifiable surface if for almost every $x \in \mathbb{M}$, there exists a tangent space $\tau_x \mathbb{M} \in \mathbb{R}^2$.

To make the abstract mathematical framework more intuitive, we consider the problem of partial matching between shapes (Antonsanti et al. 2021). Hence, the aim is to find the optimal deformation to register a source S to a target shape T, where $S, T \in \mathbb{R}^m$, hence they are a finite union of m-dimensional submanifolds of the ambient space $E = \mathbb{R}^d$. We consider them to be m-rectifiable, which in the case of hyperspheres implies that any point $x \in S$ ($x \in T$) has a normal vector $\tau_x S \in S^{d-1}$ ($\tau_x T \in \mathbb{S}^{d-1}$).

Definition A.1. A varifold is a distribution on the product $\mathbb{R}^k \times \mathbb{S}^{k-1}$, namely a continuous linear form on a given space W of smooth functions $\mathbb{R}^k \times \mathbb{S}^{k1} \to \mathbb{R}$. If M is a surface, we associate the varifold $\{M\} \in W'$, for any function $w \in W$, by the surface integral: $\{M\}(w) = \int_M w\,(x, \tau_x M)\ \mathrm{d}S(x)$, where W' is the dual space (see Chapter 3.2 in Charon et al. (2020) for an in-depth definition).

Proposition A.1 (Proposition 4.1. in Charon and Trouvé (2013))**.** *Given positive-definite real kernel k_e on $\mathbb{R}^d$ s.t. k_e is continuous and for all $x \in \mathbb{R}^d$, $k_e(x, \cdot)$ vanishes at infinity and given a positive-definite real kernel k_t defined on the manifold $\mathbb{S}^{d-1}$ that is also continuous. Then the Reproducing Kernel Hilbert Space (RKHS) W associated to the positive-definite kernel $k_e \otimes k_t$ is continuously embedded into the space $C_0\left(\mathbb{R}^d \times \mathbb{S}^{d-1}\right)$, the space of continuous functions on $\mathbb{R}^d \times$*

$\mathbb{S}^{d-1}$ *that decay to 0 at infinity.*

In Antonsanti et al. (2021), the authors assume the reproducing kernel of W *to be given by* $k_e \otimes k_t$*, with* $k_e(x,y) = \exp\left(-|x-y|^2\right)/\sigma_w^2$ *and* $k_t(u,v) = \exp\langle u,v\rangle_R^d$*, with the latter being an exponential kernel applied on the inner product of surface normals. We associate the following canonical function* $w_S \in W$*:*

$$w_S(y,\tau) = \int_S k_e(y,x)k_t(\tau,\tau_x S)\,\mathrm{d}x \tag{12}$$

We can finally define the scalar product between canonical functions w_S *and* w_T *as follows:*

$$\langle w_S, w_T\rangle = \int_S\int_T k_e(x,y)k_t(\tau_x S,\tau_y T)\,\mathrm{d}x\,\mathrm{d}y \tag{13}$$

Remark A.1. *Intuitively, the inner product associated to RKHS* W *keeps track of both Euclidean differences between 3D points on the surface mesh, but also takes into account geometric information surrounding those points, as given by the points' surface normal vectors (see Figure 3.1 in Section 3.2.2).*

A.2 Detailed proofs for Chapter 3

A.2.1 Positive-definite kernels using Wasserstein-2 distances

Proof of Theorem 3.2.1 *Thi Thien Trang et al. (2019) provides a detailed proof that a generalized radial basis function kernel with Wasserstein distances is positive definite. We recall here the main results.*

Theorem A.1 (Schoenberg's Theorem)**.** *Let* $F : \mathbb{R}^+ \Rightarrow \mathbb{R}^+$ *be strictly monotonically decreasing, and* $\mathbb{K}$ *a negative definite kernel. Then* $(x,y) \Rightarrow F(\mathbb{K}(x,y))$ *is a positive definite kernel.*

Proposition A.2. *The function* W_2^{2H} *is a negative definite kernel if and only if* $0 < H \leq 1$*.*

A complete proof of this proposition can be found in Kolouri et al. (2016), where they prove this result for absolutely continuous distributions in $W_2(\Omega)$*, where* $\Omega \subset \mathbb{R}$ *is a compact subset. Theorem 3.2.1 follows from Schoenberg's Theorem and the above proposition.*

A.2.2 Propagation of total uncertainty in DGP

Proof of Proposition 3.3.1

Proof. We consider the approximate posterior DGP as a composition of functions $q_L = q_L \circ ... \circ q_1$

with a certain q_l being given by:

$$q(F_l|F_{l-1}) = \int p(F_l|U_l, F_{l-1})q(U_l)\,\mathrm{d}U_l = \mathcal{N}(F|\tilde{U}_l(f_{l-1}), \tilde{\Sigma}_l(f_{l-1}))\,. \tag{14}$$

Following the layout introduced in Girard et al. (2002) for obtaining Gaussian approximations of GP with uncertain inputs, we have the following adapted case for a two layer DGP, with the forthcoming analysis being easily extended to more layers: U

$$q(F_2)(x) = \int p(F_2|F_1)q(F_1(x))\,\mathrm{d}F_1\,. \tag{15}$$

Using the framework described in subsection 2.5.1 (equations (2.113) and (2.116)) and adapting to our case at hand, we have the following approximate moments for $q(F_2)(x)$:

$$m(F_2(x)) = \tilde{U}_2(\tilde{U}_1(x)) \tag{16}$$

$$v(F_2(x)) = \tilde{\Sigma}_2(\tilde{U}_1(x)) + \tilde{\Sigma}_1(x)\left[\frac{1}{2}\frac{\partial^2\tilde{\Sigma}_2(F_1)}{\partial^2 F_1}\Big|_{F_1=\tilde{U}_1(x)} + \left(\frac{\partial\tilde{U}_2(F_1)}{\partial F_1}\right)^2\Big|_{F_1=\tilde{U}_1(x)}\right], \tag{17}$$

where for the purposes of our derivation here we only considered a first order Taylor expansion for the approximate mean. In Ustyuzhaninov et al. (2019), the authors have empirically shown that $\lim_{M\to\infty}\frac{\partial^2\tilde{\Sigma}_2(F_1)}{\partial^2 F_1} = 0$, hence that term is dropped. Intuitively, it follows a sinusoidal path with the amplitude converging to 0 as the number of inducing points is increased.

We remind ourselves that in the proposition statement we made the following set of assumptions: inducing point locations for the second layer Z_1 are equidistantly placed in the interval $[-3\sigma_1, 3\sigma_1]$; the variance of the inducing point values S_{U_2} are taken to be equal to simplify the problem.

For x_{out} and x_{in-d} using equation 17 we obtain:

$$v(F_2(x_{out})) = \tilde{\Sigma}_2(0) + \sigma_1^2\left(\frac{\partial\tilde{U}_2(F_1)}{\partial F_1}\right)^2\Big|_{F_1=0} \tag{18}$$

$$v(F_2(x_{in-d})) = \tilde{\Sigma}_2(M_{in-d}) + V_{in-d}\left(\frac{\partial\tilde{U}_2(F_1)}{\partial F_1}\right)^2\Big|_{F_1=M_{in-d}} \tag{19}$$

The desired behaviour for the second layer is:

$$\tilde{\Sigma}_2(M_{in-d}) + V_{in-d}\left(\frac{\partial\tilde{U}_2(x)}{\partial x}\right)^2\Big|_{F_1=M_{in-d}} \leq \tilde{\Sigma}_2(0) + \sigma_1^2\left(\frac{\partial\tilde{U}_2(x)}{\partial x}\right)^2\Big|_{F_1=0}. \tag{20}$$

Since we have assumed that the inducing point locations are equidistantly placed in a grid with centre zero, which is a sufficiently similar situation to what one might observe in practice, then

$\tilde{\Sigma}_2(0) \approx \tilde{\Sigma}_2(M_{in-d})$ as we have also assumed that the posterior variances of the inducing point values are equal. Hence, the inequality $v(F_2(x_{in-d})) \leq v(F_2(x_{out}))$ holds if and only if:

$$\frac{\left(\frac{\partial \tilde{U}_2(F_1)}{\partial F_1}\right)^2\Big|_{F_1=M_{in-d}}}{\left(\frac{\partial \tilde{U}_2(F_1)}{\partial F_1}\right)^2\Big|_{F_1=0}} \leq \frac{\sigma_1^2}{V_{in-d}} \,. \tag{21}$$

□

A.2.3 Propagation of distributional uncertainty in DGP

Proof of Proposition 3.3.2

Proof. We commence by writing our Bayesian hierarchical formulation so that the link with approximation of Gaussian Processes' moments with noisy inputs (see section 2.5.1) is made clearer. Under our set of assumptions we have:

$$p(H_1 \mid H_0) = \mathcal{N}\left(H_1 \mid 0, K_{ff} - K_{fu}K_{uu}^{-1}K_{uf}\right) \tag{22}$$

$$= \mathcal{N}\left(H_1 \mid 0, \sigma_1^2\right) \tag{23}$$

$$p(H_2 \mid H_1) = \int \mathcal{N}\left(H_2 \mid 0, K_{ff} - K_{fu}K_{uu}^{-1}K_{uf}\right)\mathcal{N}\left(H_1 \mid 0, \sigma_1^2\right)\, \mathrm{d}H_1 \,. \tag{24}$$

We can approximate the non-Gaussian predictive equation using the framework of exact moments introduced in subsection 2.5.2 since we are only considering squared exponential kernels. We adapt equation (2.133) to suit our scenario:

$$v\left(0, \sigma_1^2\right) = \sigma_2^2 - \mathrm{Tr}\left[K_{uu}^{-1}Q\right] \,, \tag{25}$$

where we have:

$$K = \begin{pmatrix} \sigma_2^2 & \sigma_2^2 \exp\left[-\frac{2c^2}{l^2}\right] \\ \sigma_2^2 \exp\left[-\frac{2c^2}{l^2}\right] & \sigma_2^2 \end{pmatrix} \tag{26}$$

$$Q = \frac{1}{\sqrt{\frac{2\sigma_1^2}{l^2}+1}} \begin{pmatrix} \exp\left[-\frac{c^2}{l^2+2\sigma_1^2}\right] & \exp\left[-\frac{c^2}{l_2}\right] \\ \exp\left[-\frac{c^2}{l_2}\right] & \exp\left[-\frac{c^2}{l^2+2\sigma_1^2}\right] \end{pmatrix} \,. \tag{27}$$

We can now write the exact variance of the Gaussian approximation of $p(H_2)$ as:

$$v\left(0, \sigma_1^2\right) = \sigma_2^2 - \frac{2\left[\exp\left[-\frac{c^2}{l^2+2\sigma_1^2}\right] - \exp\left[-\frac{3c^2}{l^2}\right]\right]}{\sigma_2^2\left[1 - \exp\left(-\frac{4c^2}{l^2}\right)\right]\sqrt{\frac{2\sigma_1^2}{l_2}+1}} \,, \tag{28}$$

which has the following derivative with respect to σ_1^2:

$$\frac{\partial v\left(0,\sigma_1^2\right)}{\partial \sigma_1^2} = \sigma_2^2\left[1-\exp\left(-\frac{4c^2}{l^2}\right)\right] 2\exp\left[-\frac{c^2}{l^2+2\sigma_1^2}\right]\left[\frac{2\sigma_1^2+l^2}{l^2}\right]^{-\frac{1}{2}} \left[2c^2\left[l^2+2\sigma_{l-1}^2\right]-\frac{1}{2\sigma_1^2+l^2}\right] . \tag{29}$$

We first consider the derivative evaluated at $\sigma_1^2 = 0$:

$$\frac{\partial v\left(0,\sigma_1^2\right)}{\partial \sigma_1^2}\Big|_{\sigma_1^2=0} = \sigma_2^2\left[1-\exp\left(-\frac{4c^2}{l^2}\right)\right]\left[-\frac{c^2}{l^2}\right]\left[\frac{2c^2}{l^4}-\frac{1}{l^2}\right] . \tag{30}$$

The first two terms are always positive in the case that K_{uu} is invertible, leaving us with just the last term. Hence, we have that:

$$\frac{\partial v\left(0,\sigma_1^2\right)}{\partial \sigma_1^2}\Big|_{\sigma_1^2=0} \geq 0 \iff l^2 \geq 2c^2 . \tag{31}$$

Similarly, one can investigate whether the derivative is positive at any other value except 0:

$$\frac{\partial v\left(0,\sigma_1^2\right)}{\partial \sigma_1^2}\Big|_{\sigma_1^2=\tilde{\sigma}_1^2} \geq 0 \iff \tilde{\sigma}_1^2 + l^2 \geq 2c^2 . \tag{32}$$

One can easily see that at any value of σ_1^2 the derivative will be positive only if $l^2 \geq 2c^2$. Hence, the minimum values will be at $v(0,0) = K_{ff} - Q_{ff}$, where the Schur complement is evaluated at point $F_1 = 0$. □

A.2.4 Propagation of total uncertainty in DDGP

Proof of Proposition 3.4.1

Proof. From the proof of Proposition 3.3.1 we remind the desired behaviour for the second layer:

$$\tilde{\Sigma}_2(M_{in-d}) + V_{in-d} * \left(\frac{\partial \tilde{U}_2(F_1)}{\partial F_1}\right)^2\Big|_{F_1=M_{in-d}} \leq \tilde{\Sigma}_2(0) + \sigma^2 * \left(\frac{\partial \tilde{U}_2(F_1)}{\partial F_1}\right)^2\Big|_{F_1=0} \tag{33}$$

In contrast to zero mean DGP, where under current assumptions we obtained $\tilde{\Sigma}_2(M_{in-d}) \approx \tilde{\Sigma}_2(0)$, this is not the case for DDGP. Expanding on the variance terms $\tilde{\Sigma}_2(M_{in-d})$ and $\tilde{\Sigma}_2(0)$ looking at their non-parametric variance:

$$\sigma_2^2 - K_{M_{in-d}u}^H {K_{uu}^H}^{-1} K_{uM_{in-d}}^H \leq \sigma_2^2 - K_{0u}^H {K_{uu}^H}^{-1} K_{u0}^H . \tag{34}$$

To simplify the notation we can consider just one inducing point, lying halfway between 0 and M_{in-d} at Z_m and with $Z_{var} \ll \sigma_1^2$. Using the hybrid kernel formulation and re-arranging the terms

in the above equation considering the matrices are scalars we arrive at:

$$\sigma_2^2 \exp \frac{-W_2^2(\mathcal{N}(0,\sigma_1^2),\mathcal{N}(Z_m,Z_{var}))}{l_2^2} \exp \frac{-(0-Z_m)^2}{l_2^2} \leq \\ \sigma_2^2 \exp \frac{-W_2^2(\mathcal{N}(M_{in-d},V_{in-d}),\mathcal{N}(Z_m,Z_{var}))}{l_2^2} \exp \frac{-(M_{in-d}-Z_m)^2}{l_2^2}\,, \tag{35}$$

which after eliminating redundant terms and taking the logarithm we obtain:

$$W_2^2(\mathcal{N}(0,\sigma_1^2),\mathcal{N}(Z_m,Z_{var})) + (0-Z_m)^2 \geq W_2^2(\mathcal{N}(M_{in-d},V_{in-d}),\,\mathcal{N}(Z_m,Z_{var}))+ \\ (M_{in-d}-Z_m)^2 \tag{36}$$

The above inequality holds if and only if $\sigma_1^2 \gg Z_{var}$ and $V_{in-d} \approx Z_{var}$, thus proving our proposition.

□

A.2.5 Feature-collapse in DGP

Proof of Proposition 3.4.2

Proof. In subsequent derivations, we assume that the set of inducing points $\{Z_l\}_{l=1}^L$ are sufficiently large and well-optimized so that for points $x \in D$ we have $K_{ff} \approx K_{fu}K_{uu}^{-1}K_{uf}$.

We commence the derivation:

$$\mathbb{E}\left[\|f_n(x) - f_n(x^*)\|_2^2 \mid f_{n-1}, U_n\right] = \sum_{d=1}^{D_l} \mathbb{E}\left[\left|f_n^d(x) - f_n^d(x^*)\right|^2 \mid f_{n-1}, U_n\right] \tag{37}$$

$$= \sum_{d=1}^{D_l} \left|\left(K_{fu} - K_{f^*u}\right) K_{uu}^{-1} U_n^d\right|^2 \tag{38}$$

We can apply Cauchy-Schwarz inside the brackets to obtain the following inequality:

$$\leq \sum_{d=1}^{D_l} \|K_{fu} - K_{f^*u}\|_2^2 \|K_{uu}^{-1}U_n^d\|_2^2 \tag{39}$$

$$\leq \sum_{d=1}^{D_l} \|\sigma_n^2 \exp -\frac{(x-Z_n)^2}{2l_n^2} - \sigma_n^2 \exp -\frac{(x^*-Z_n)^2}{2l_n^2}\|_2^2 \|K_{uu}^{-1}U_n^d\|_2^2\,. \tag{40}$$

We assume that $x = x^* + h$ and x^* is a high-density data point with regards to Z_n, respectively

$|| \, x - Z_n \, ||_2^2 \approx 0$. Now we can re-arrange the last equation as:

$$\leq \sum_{d=1}^{D_l} \|\sigma_n^2 \exp - \frac{(h + x^* - Z_n)^2}{2l_n^2} - \sigma_n^2\|_2^2 \|K_{uu}^{-1} U_n^d\|_2^2 \tag{41}$$

$$\leq \sum_{d=1}^{D_l} \sigma_n^4 \mid \exp - \frac{(h)^2}{2l_n^2} - 1 \mid^2 \|K_{uu}^{-1} U_n^d\|_2^2 \tag{42}$$

We have the following general result $1 - \exp -x < x, \text{ for } x > -1$ which we apply in our case since $\frac{(h)^2}{2l^2} > -1$, we obtain:

$$\mathbb{E}\left[\|f_n(x) - f_n(x^*)\|_2^2 \mid f_{n-1}, U_n\right] \leq \sum_{d=1}^{D_l} \frac{\sigma_n^4}{4l_n^4} \|K_{uu}^{-1} U_n^d\|_2^2 \, (h)^2 \, (h)^2 \ , \tag{43}$$

which we can re-express with regards to previous layer features to make subsequent derivations easier to follow:

$$\mathbb{E}\left[\|f_n(x) - f_n(x^*)\|_2^2 \mid f_{n-1}, U_n\right] \leq \sum_{d=1}^{D_l} \frac{\sigma_n^4}{4l_n^4} \|K_{uu}^{-1} U_n^d\|_2^2 \, (h)^2 \, (f_{n-1}(x) - f_{n-1}(x^*))^2 \ . \tag{44}$$

We can now apply induction and the tower property of conditional expectation to arrive at:

$$\mathbb{E}\left[\|f_n(x) - f_n(x^*)\|_2^2 \mid \{U_l\}_{l=1}^n\right] \leq \prod_{l=1}^{n} \frac{\sigma_l^4}{4l_l^4} \|K_{uu}^{-1} U_l\|_2^2 \, (h)^2 \, (f_0(x) - f_0(x^*))^2 \ . \tag{45}$$

By Markov's inequality, for any $\epsilon > 0$ we have that:

$$P\left(\|f_{l+1}(x) - f_{l+1}(x^*)\|_2 \geq \epsilon \mid \{U_l\}_{l=1}^n\right) \leq \frac{1}{\epsilon^2} \prod_{l=1}^{n} \frac{\sigma_l^4}{4l_l^4} \|K_{uu}^{-1} U_l\|_2^2 \, (h)^2 \, (f_0(x) - f_0(x^*))^2 \ . \tag{46}$$

We can apply the first Borel-Cantelli lemma to obtain:

$$P\left(\cap_{l=1}^{\infty} \cup_{m=l}^{\infty} \|f_m(x) - f_m(x^*)\|_2 \geq \epsilon \mid \{U_l\}_{l=1}^n\right) = 0 \, . \tag{47}$$

Lastly, we can express the following:

$$
\begin{aligned}
P\left(\|f_n(x) - f_n(x^*)\|_2 \to 0 \mid \{U_l\}_{l=1}^n\right) &= P(\cap_{k=1}^{\infty} \cup_{l=1}^{\infty} \cap_{m=l}^{\infty} \|f_m(x) - f_m(x^*)\|_2 && (48)\\
&\quad \leq \frac{1}{k} \mid \{U_l\}_{l=1}^n) \\
&= 1 - P(\cup_{k=1}^{\infty} \cap_{l=1}^{\infty} \cup_{m=l}^{\infty} \|f_m(x) - f_m(x^*)\|_2 && (49)\\
&\quad \geq \frac{1}{k} \mid \{U_l\}_{l=1}^n) \\
&\geq 1 - \sum_{k=1}^{\infty} P(\cap_{l=1}^{\infty} \cup_{m=l}^{\infty} \|f_m(x) - f_m(x^*)\|_2 && (50)\\
&\quad \geq \frac{1}{k} \mid \{U_l\}_{l=1}^n) \\
&= 1\,. && (51)
\end{aligned}
$$

From which we obtain the proof of our proposition: $P\left(\|f_n(x) - f_n(x^*)\|_2 \to 0 \mid \{U_l\}_{l=1}^n\right) = 1$. □

A.2.6 Feature-collapse in DDGP

Proof of Proposition 3.4.3

Proof. We adapt our previous proof for Proposition 3.4.2 to suit the hybrid kernel specific to DDGP. We have the following upper bound:

$$
\begin{aligned}
\mathbb{E}\left[\|f_n(x) - f_n(x^*)\|_2^2 \mid f_{n-1}, U_n\right] \leq \sum_{d=1}^{D_n} \|\sigma_n^2 \exp -\frac{(x - Z_n)^2 + W_2^2(\mu(x), \mu(Z_n))}{2l_n^2} \\
- \sigma_n^2 \exp -\frac{(x^* - Z_n)^2 + W_2^2(\mu(x^*), \mu(Z_n))}{2l_n^2} \|_2^2 \\
\|K_{uu}^{-1} U_n^d\|_2^2
\end{aligned} \tag{52}
$$

We assume that $x = x^* + h$ and x^* is a high-density data point with regards to Z_n, respectively $||\, x - Z_n\,||_2^2 \approx 0$. For their first two moments we have equivalent assumptions $m(x) = m(x^*) + m(h)$ and $v(x) = v(x^*) + v(h)$, with $\mu(x^*) \overset{d}{\approx} \mu(Z_n)$ We can re-arrange the last equation as:

$$
\leq \sum_{d=1}^{D_n} \|\sigma_n^2 \exp -\frac{(h + x^* - Z_n)^2 + \|m(h)\|_2^2 + \|v(h)\|_2^2}{2l_n^2} - \sigma^2\|_2^2 \|K_{uu}^{-1} U_n^d\|_2^2 \tag{53}
$$

We have the following general result $1 - \exp -x < x,$ for $x > -1$ which we apply in our case since $\frac{(h)^2 + |m(h)|^2 + |v(h)|^2}{2l^2} > -1$:

$$
\mathbb{E}\left[\|f_n(x) - f_n(x^*)\|_2^2 \mid f_{n-1}, U_n\right] \leq \sum_{d=1}^{D_n} \frac{\sigma_n^4}{4l_n^4} \|K_{uu}^{-1} U_n^d\|_2^2 \left(h^2 + \|m(h)\|_2^2 + \|v(h)\|_2^2\right)^2 \tag{54}
$$

We can see that we can re-express the final term of above equation as follows:

$$\left(h^2 + \underbrace{\|m(h)\|_2^2 + \|v(h)\|_2^2}_{c}\right)^2 = \left(h^2 + c\right)^2 \tag{55}$$

$$= h^4 + 2h^2c + c^2 \tag{56}$$

$$= h^2\left[h^2 + 2c + \frac{c^2}{h^2}\right] \tag{57}$$

$$= h^2\left[h + \frac{c}{h}\right]^2 \tag{58}$$

We can apply this side derivation to equation (54):

$$\mathbb{E}\left[\|f_n(x) - f_n(x^*)\|_2^2 \mid f_{n-1}, U_n\right] \leq \sum_{d=1}^{D_n} \frac{\sigma_n^4}{4l_n^4}\|K_{uu}^{-1}U_n^d\|_2^2 h^2 \left[h + \frac{\|m(h)\|_2^2 + \|v(h)\|_2^2}{h}\right]^2 h^2 \tag{59}$$

We can now apply induction and the tower property of conditional expectation to arrive at:

$$\mathbb{E}\left[\|f_n(x) - f_n(x^*)\|_2^2 \mid \{U_l\}_{l=2}^n\right] \leq \prod_{l=2}^{L} \frac{\sigma_l^4}{4l_l^4}\|K_{uu}^{-1}U_l^d\|_2^2 \left[h + \frac{\|m(h)\|_2^2 + \|v(h)\|_2^2}{h}\right]^2 \tag{60}$$
$$\left(f_1(x) - f_1(x^*)\right)^2$$

Using the upper bound on $\left(f_1(x) - f_1(x^*)\right)^2$ from the proof of Proposition 3.4.2, we obtain:

$$\mathbb{E}\left[\|f_n(x) - f_n(x^*)\|_2^2 \mid \{U_l\}_{l=1}^n\right] \leq \frac{\sigma_1^4}{4l_1^4}\|K_{uu}^{-1}U_1^d\|_2^2 \prod_{l=2}^{L} \frac{\sigma_l^4}{4l_l^4}\|K_{uu}^{-1}U_l^d\|_2^2 \tag{61}$$
$$\left[h + \frac{\|m(h)\|_2^2 + \|v(h)\|_2^2}{h}\right]^2$$
$$\left(f_0(x) - f_0(x^*)\right)^2$$

Similar derivations to the proof of Proposition 3.4.2 can be applied to arrive at the final result. □

A.3 UCI experiments details

For all datasets we randomly selected 20% as the testing set, with the remainder being used for training. All implementations use the RBF kernel with automatic relevance determination for DGP, whereas for DDGP the standard RBF kernel on Euclidean space is used just for the first layer, with the remaining layers using the Hybrid kernel. In terms of model architecture, we use 5 hidden units for each hidden level, with 100 inducing inputs. All models are optimized until convergence with a mini-batch of size 32. Results are provided for 2,3 and 4 layers.

Euclidean Kernels. *All squared exponential kernels used a lengthscale hyperparameter per input dimension, initialized to 1.351. We initialize the variance of the kernel with the same value.*

Hybrid Kernels. *All hybrid kernels used a lengthscale hyperparameter per input dimension, initialized to 1.351. We initialize the variance of the kernel with the same value.*

Euclidean inducing points & DGP approximate posterior. *For all experiments we used 100 inducing points for the first layer, which is mapping from input space to first hidden layer space. We initialize the inducing point locations to the k-means of the training data. For the approximate posterior of the inducing point values we initialized the mean by uniformly sampling between -2.0 and 2.0. The Cholesky decomposition of the variance was initialized to be the diagonal matrix multiplied by* $1e^{-5}$.

Distributional inducing points & DDGP approximate posterior. *For all experiments we used 100 inducing points in the hidden layers of the multi-layered GP architectures. Since these inducing point are operating in Wasserstein space, the inducing point locations are actually distributions in this case. We initialize the mean of the inducing points locations by uniformly sampling between -2.0 and 2.0. The initial variance of the inducing point locations is taken to be the diagonal matrix multiplied by* $1e^{-1}$. *This low initialization of the variance term was chosen so that the DDGP will implicitly learn to produce low variance terms at all hidden layers for data points inside the data manifold. For the approximate posterior of the inducing point values, we take the same initialization as in the case of DGP.*

Optimization. *All parameters were optimized using the Adam optimizer with a learning rate of 0.001. We used a mini batch size of 32 and trained for 1000 epochs or until convergence.*

Likelihood. *For regression tasks the likelihood variance was initialized to 1.0*

A.4 Isotropic kernel properties in (Distributional) Deep Gaussian Processes

Point-mass inducing points are doomed to fail in high-dimensional spaces. *This section is written from a purely theoretical perspective in the hypothetical scenario that DeepConvGP architectures or more generally DeepGP architectures are scalable to high dimensions. Operating in this scenario, we can take the case of DGP, with the following analysis being easily extendable to DeepConvGP architectures. We first introduce a general result from the concentration of measure in high-dimensional spaces.*

Definition A.2. Multivariate spherical Gaussian distributions with a different mean u_k and variance σ_k^2 for each dimension has the following p.d.f:

$$p(x) = \frac{1}{(2\pi)^{d/2}\prod_{k=1}^{K}\sigma_k}\exp\left[-\sum_{k=1}^{K}\left(\frac{(x_k-u_k)^2}{2\sigma_k^2}\right)\right] \tag{62}$$

Definition A.3 (Gaussian Annulus Theorem)**.** For a d-dimensional spherical Gaussian with unit variance in each direction, we have that:

$$Pr(|x-\sqrt{d}| \leq \beta) \leq 3\exp -c\beta^2 \tag{63}$$

where c is a fixed constant. This results in samples being concentrated in a thin annulus of size 1 with a radius $\sqrt{d}$, centred at zero.

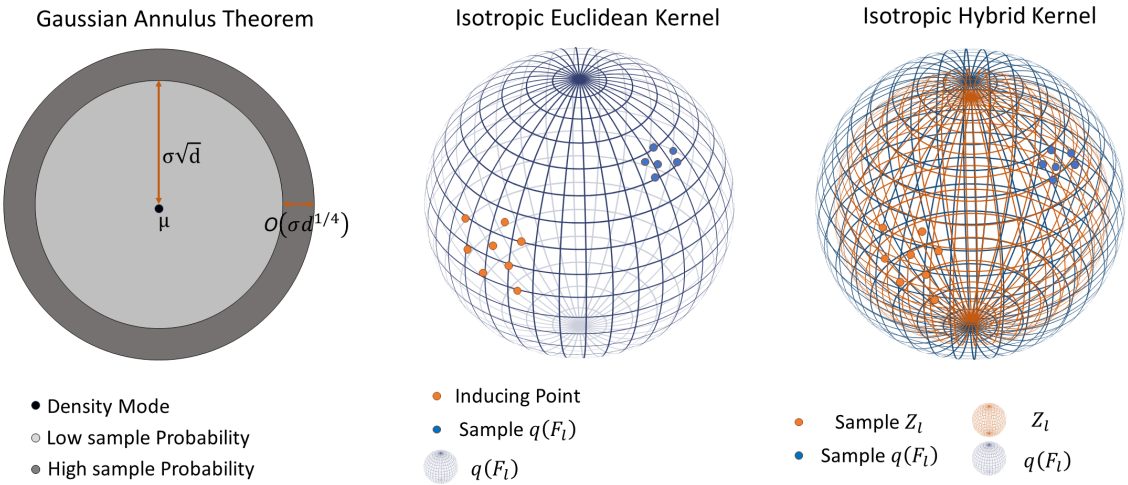


Figure A.1: Gaussian Annulus Theorem and behaviour of (D)DGP in high-dimensional spaces. Left: Visual depiction of Gaussian Annulus Theorem; Middle: Hidden layer samples in DGP concentrate on a thin hypersphere, with inducing points having to cover the entire surface; Right: Hidden layer samples in DDGP concentrate on a thin hypersphere, but due to distributional inducing points and Wasserstein-2 part of hybrid kernel this will not result in the necessity to cover entire hypersphere.

If we consider the posterior over the l-th layer of a DGP $q(F_l) \sim \mathcal{N}\left(F_l; \tilde{U}_l, \tilde{\Sigma}_l\right)$ *be of size* d_l *where we consider the dimensionality to be sufficiently large so that the above theorem holds, with the added assumption that the d-dimensional* $\tilde{\Sigma}_l$ *values for data point* x *are roughly equal. Then we can consider samples from* $q(F_l)$ *as* $\{F_l^1 - \tilde{U}_l, \cdots, F_l^S - \tilde{U}_l\}$*, which are individually spherical Gaussians of mean zero and* $\tilde{\Sigma}_l$ *variance. Thereby, by applying the Gaussian Annulus Theorem we can see that samples, which are used in the DSVI framework commonly used in DGP (Salimbeni and Deisenroth 2017), concentrate around an annulus centred at* $\tilde{U}_l$ *with the radius being* $\sqrt{d_l\Sigma_l}$*. Hence, the Euclidean distance with respect to the centre increases for higher values of* d_l*.*

Remark A.2. *This will only constitute an issue with isotropic kernels, as the locations* Z_l *will have*

to cover an increasing surface of the d_l-sphere created by samples from $q(F_l)$. This will constitute less of a problem for hybrid kernels (isotropic as well for Euclidean part) present in DDGP, as the Wasserstein-2 distance between the multivariate normal stemming from $q(F_l)$ and that of Z_l will be reliable.

A.5 Additional Results on Out-of-Distribution Detection for DDGP

Deep Convolutional Gaussian Processes architecture. *We use 250 inducing points at each layer. Every hidden layer is taken to have five channels. We use the Adam optimizer starting with a learning rate of 0.01 and decreasing by 0.1 every 20,000 iterations. We optimized all models trained on MNIST for 100,000 iterations.*

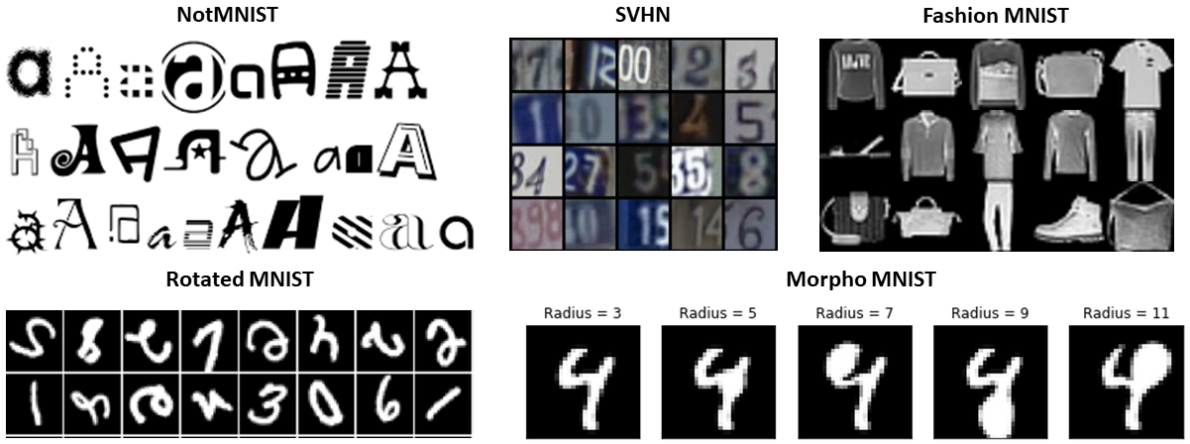


Figure A.2: Examples of OOD datasets.

Image datasets. *MNIST is a large database of small, square 28×28 pixel grayscale images of handwritten single digits between 0 and 9. It consists of 60,000 images in the training set and 10,000 in the testing set.*

NotMNIST is a dataset consisting of publicly available fonts and glyphs extracted from them so as to construct a dataset similar to MNIST. There are 10 classes, with letters A-J taken from different fonts.

SVHN is a real-world image dataset obtained from house numbers in Google Street View images with the purpose of recognizing digits and numbers in natural scene images.

Visual examples of OOD datasets used throughout thesis are given in Figure A.2.

A.5.1 Sensitivity to input perturbation

Rotated-MNIST. *To further assess the sensitivity of our methods, we employ the experiments introduced in Gal and Ghahramani 2016b by successively rotating digits from MNIST. We expect to*

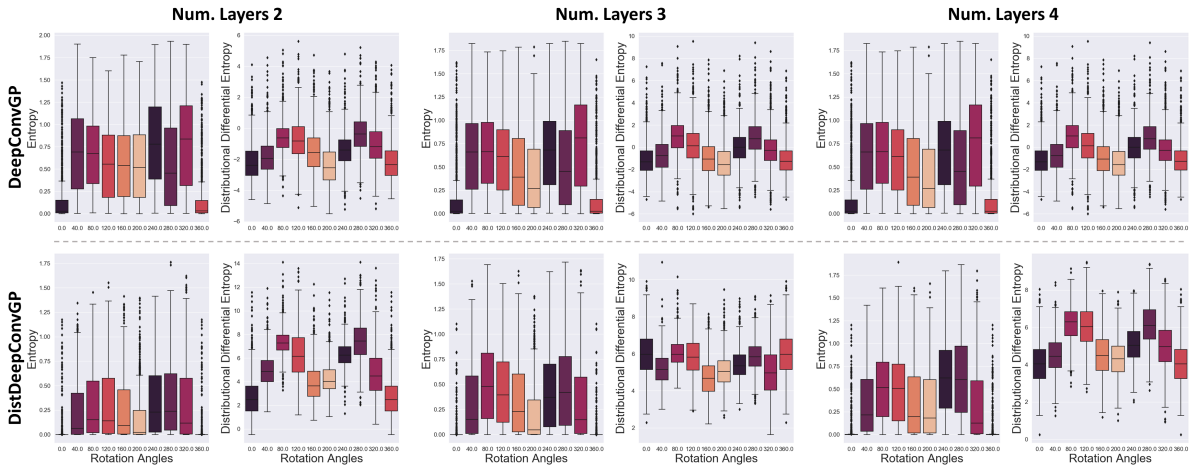


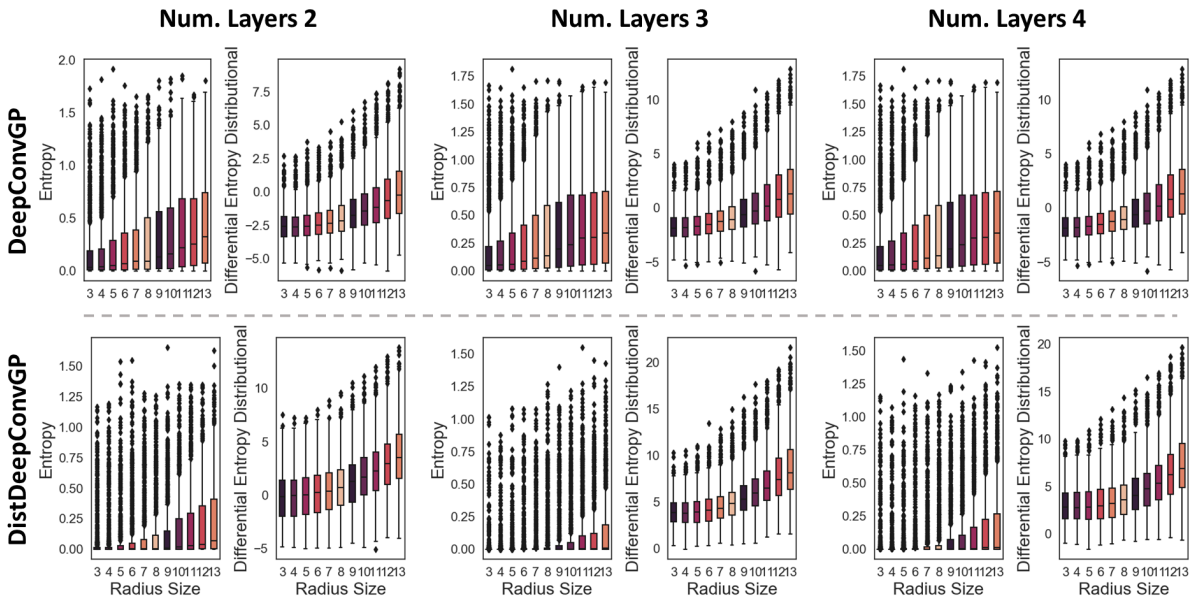
Figure A.3: Sensitivity to image rotation. Histograms of predictive entropy and distributional differential entropy for varying degrees of rotation applied to digit 8 images from MNIST. Higher values indicated an increased uncertainty. Models trained use 50 inducing points in the hidden layers.

see an increase in both predictive entropy and distributional differential entropy as digits are rotated. For our experiment we rotate digit 6. When the digit is rotated by around 180 degrees the predictive entropy and distributional differential entropy should revert back closer to initial levels, as it will resembles digit 9. From Figure A.3 we can notice that in terms of distributional differential entropy the same patterns are exhibited by both considered models, with slightly less overlapping distributions for DistDeepConvGP.

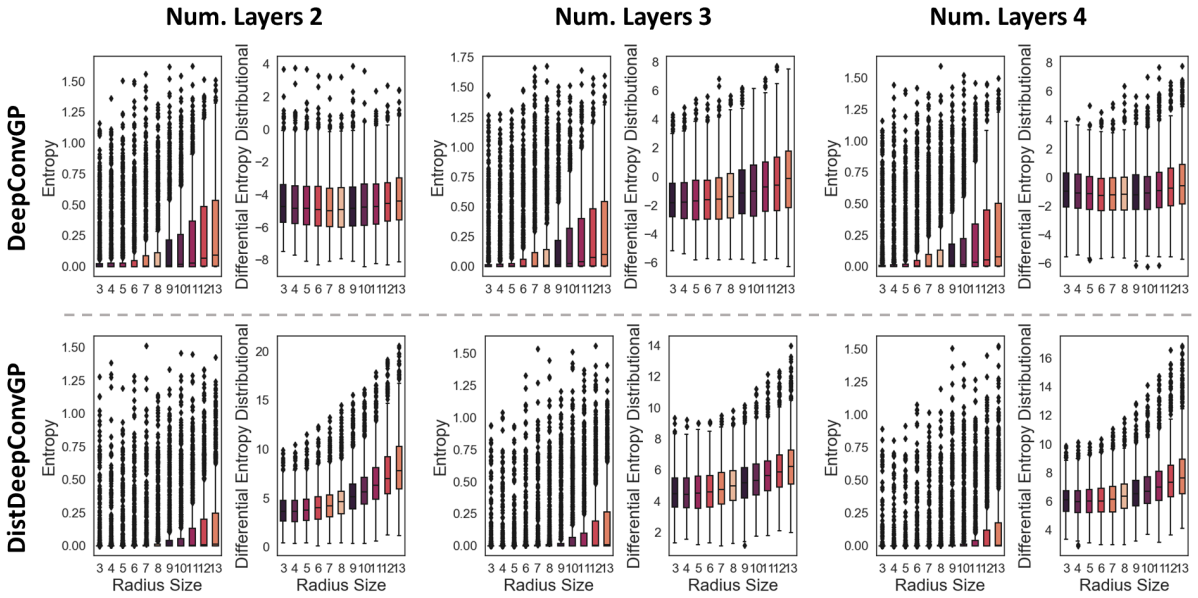
Morpho-MNIST. *For better understanding the morphological changes occurring with swelling of different radii we provide some examples for digit 4 (see Figure A.2). From Figure A.4 we can see that for 50 inducing points at each hidden layer, DGP and DDGP have roughly similar behaviour. However, for 100 inducing points we can notice that DGP (i.e., for 2 and 4 layers) lose their ability to notice these morphological changes in the input by virtue of inspecting their distributional differential entropy. The same cannot be said for DDGP, which regardless of inducing point numbers provide a steady upward trend in distributional differential entropy as swellings of higher intensity are applied on the original digit. This experiment goes hand in hand with our observation on toy datasets (see Figure 3.6), where we have seen that by increasing the number of inducing points the distributional uncertainty in zero-mean function DGP can collapse.*

A.5.2 Additional Figures for OOD detection

In this subsection we provide additional figures pertaining to the OOD detection experiments summarized in Table 3.2.

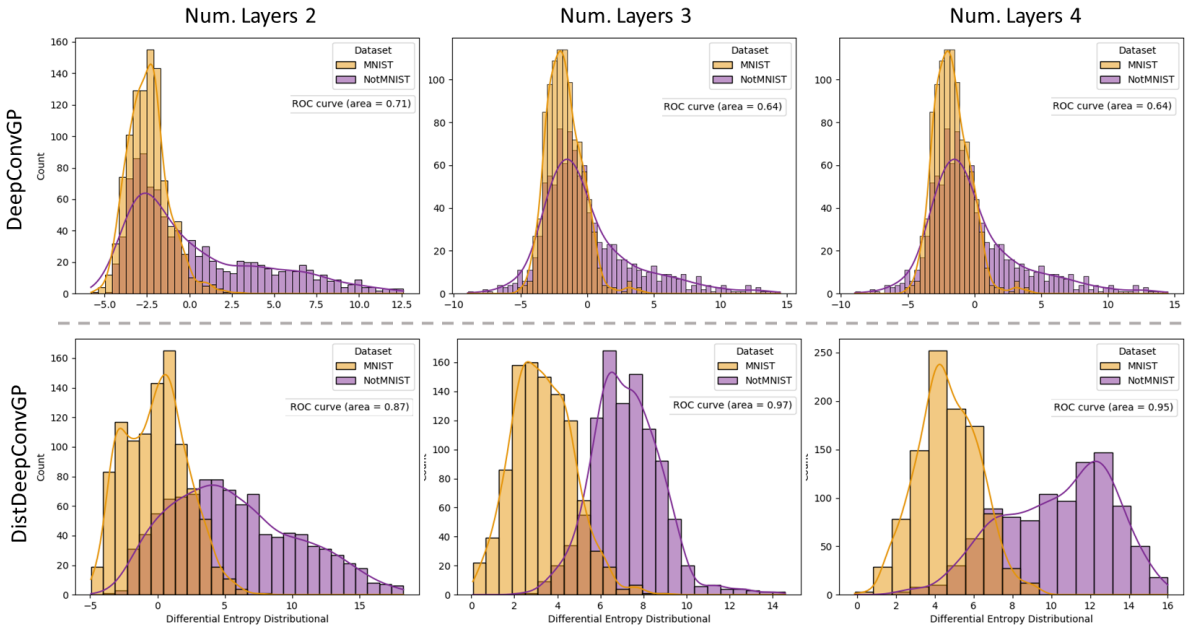


(a) 50 inducing points

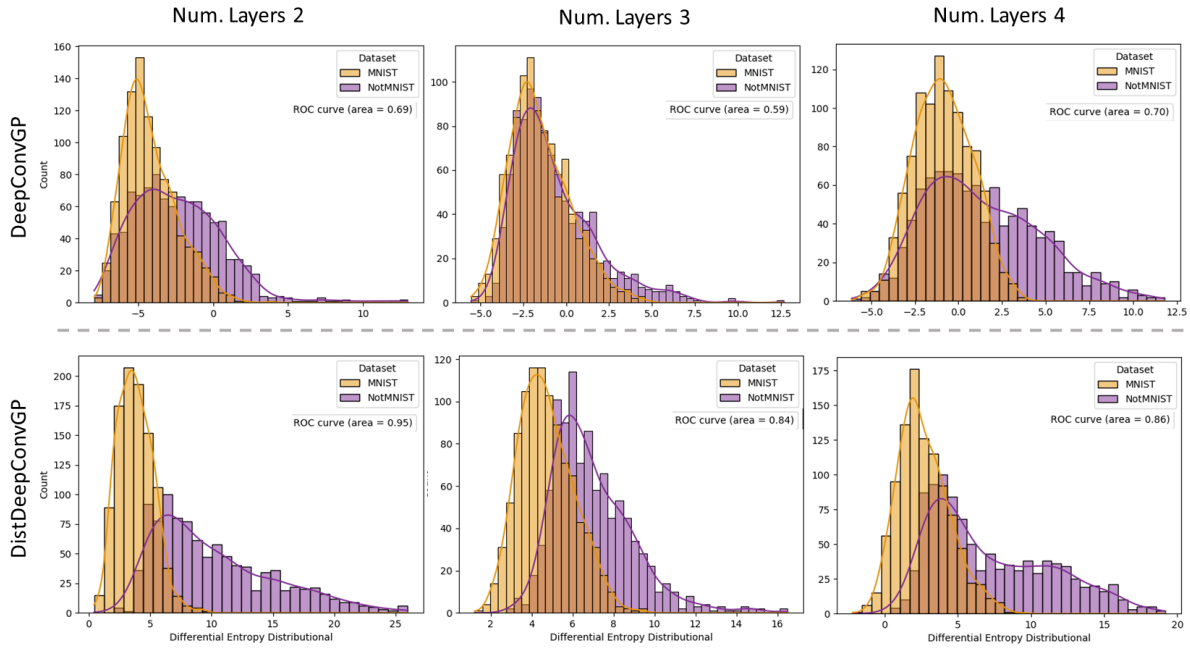


(b) 100 inducing points

Figure A.4: Sensitivity to morphological deformations. Histograms of predictive entropy and distributional differential entropy for varying degrees of morphological deformation applied on MNIST digits. Higher values indicated an increased uncertainty.

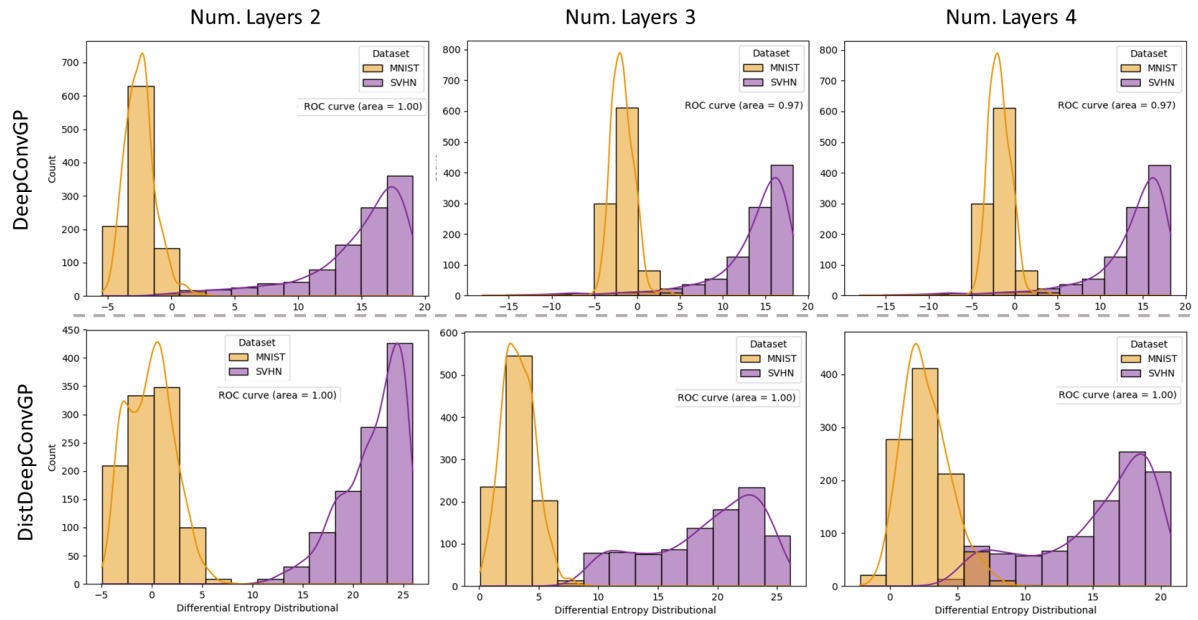


(a) 50 inducing points

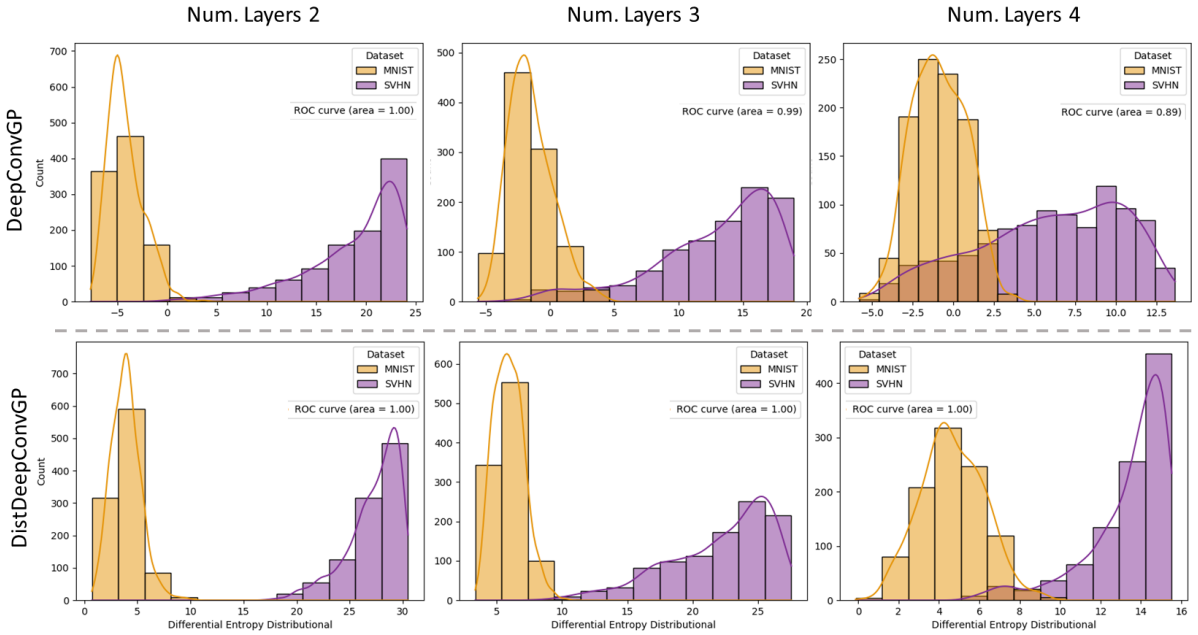


(b) 100 inducing points

Figure A.5: OOD detection MNIST vs. NotMNIST. Histograms of distributional differential entropy computed on MNIST and NotMNIST. Higher values indicate an increased uncertainty.

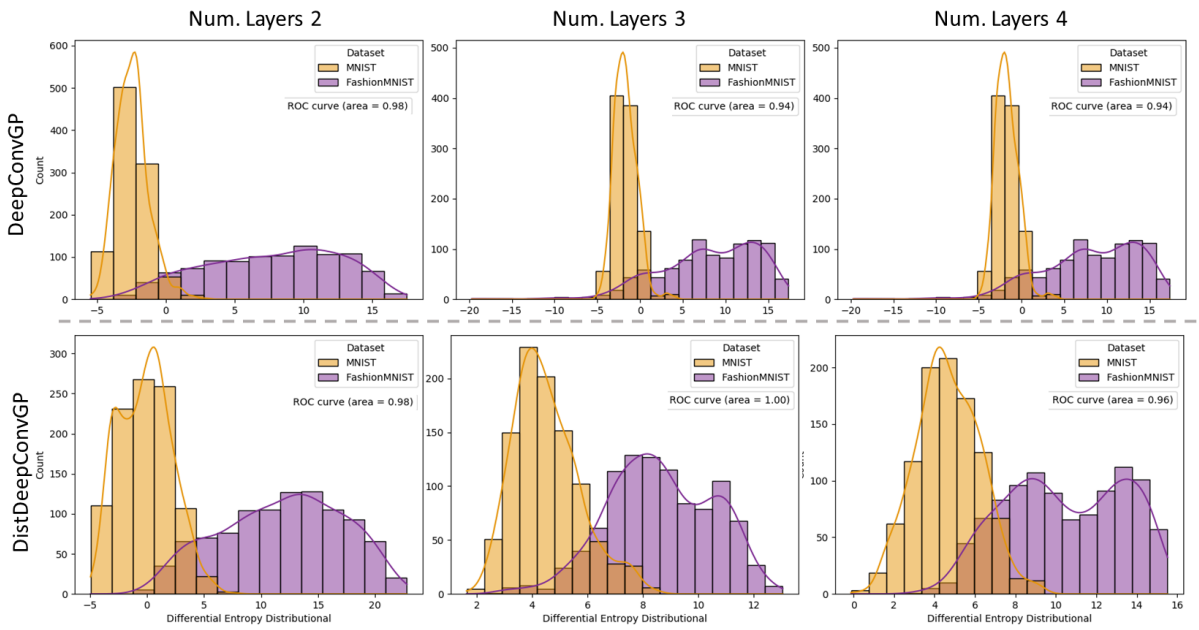


(a) 50 inducing points

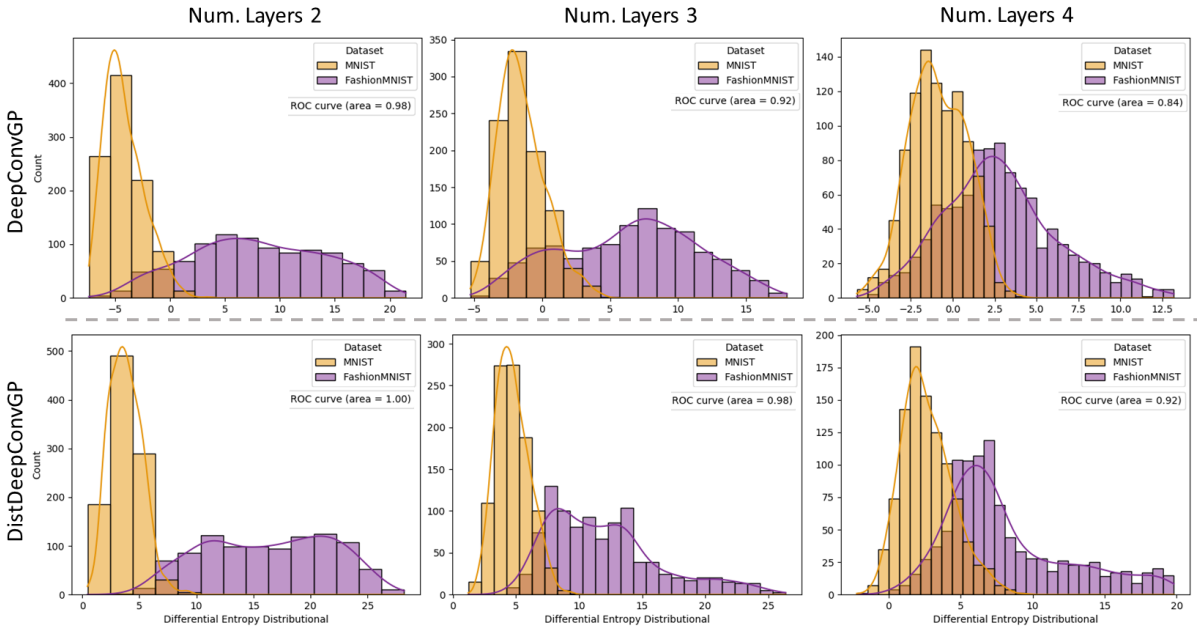


(b) 100 inducing points

Figure A.6: OOD detection MNIST vs. SVHN. Histograms of distributional differential entropy computed on MNIST and SVHN. Higher values indicate an increased uncertainty.



(a) 50 inducing points



(b) 100 inducing points

Figure A.7: OOD detection MNIST vs. FashionMNIST. Histograms of distributional differential entropy computed on MNIST and FashionMNIST. Higher values indicate an increased uncertainty.

B Further details pertaining to Distributional GP Layers

B.1 Detailed proofs for Chapter 4

B.1.1 Lipschitz properties of DistGP Layers

Proof of Proposition 4.3.1

Proof. Throughout this subsection we shall refer to the first two moments of a Gaussian distribution by $m(\cdot)$, $v(\cdot)$. Explicitly writing the Wasserstein-2 distances of the inequality we get:

$$|m(F(\mu)) - m(F(\nu))|^2 + |v(F(\mu)) - v(F(\nu)|^2 \leq L\,|m_1 - m_2|^2 + |\Sigma_1 - \Sigma_2|^2 \ . \tag{64}$$

We focus on the mean part and applying Cauchy–Schwarz we get the following inequality:

$$\left|[K_{\mu u} - K_{\nu u}]\, K_{uu}^{-1} m\right|^2 \leq \|K_{\mu u} - K_{\nu u}\|_2^2 \|K_{uu}^{-1} m\|_2^2 \,. \tag{65}$$

To simplify the problem and without loss of generality we consider U_z to be a sufficient statistic for the set of inducing points Z. Expanding the first term of the r.h.s. we get:

$$\left[\sigma^2 \exp \frac{-W_2(\mu, U_z)}{l^2} - \sigma^2 \exp \frac{-W_2(\nu, U_z)}{l^2}\right]^2 \ . \tag{66}$$

We assume $\nu = \mu + h$, where $h \sim \mathcal{N}(|m_1 - m_2|\,, |\Sigma_1 - \Sigma_2|)$ and μ is a high density point in the data manifold, hence $W_2(\mu - U_z) = 0$. We denote $m(h)^2 + var(h)^2 = \lambda$. Considering the general equality $\log(x - y) = \log(x) + \log(y) + \log(\frac{1}{y} - \frac{1}{x})$ and applying it to our case we get that:

$$\log |m(F(\mu)) - m(F(\nu))|^2 \leq \log \left[\sigma^2 - \sigma^2 \exp \frac{-\lambda}{l^2}\right]^2 \tag{67}$$

$$\leq 2 \log \sigma^2 - 2\frac{\lambda}{l^2} + 2 \log \left[\exp \frac{\lambda}{l^2} - 1\right] \tag{68}$$

$$\leq 2 \log \left[\sigma^2 \exp \frac{\lambda}{l^2}\right] \ . \tag{69}$$

We have the general inequality $\exp x \leq 1 + x + x^2$ for $x \leq 1.79$, which for $0 \leq x \leq 1$ can be modified as $\exp x \leq 1 + 2x$. Applying this new inequality and taking the exponential we now

obtain:

$$|m(F(\mu)) - m(F(\nu))|^2 \leq \left[\sigma^2 + 2\sigma^2\frac{\lambda}{l^2}\right]^2 \tag{70}$$

$$\leq \sigma^4 + \sigma^4\frac{\lambda}{l^2} + 4\sigma^4\frac{(\lambda)^2}{l^4} \tag{71}$$

$$\leq 16\sigma^4\frac{\lambda}{l^2}\,, \tag{72}$$

where the last inequality follows from the ball constraints made in the definition. We now move to the variance components of the Lipschitz bound, we notice that

$$\left|v(F(\mu))^{\frac{1}{2}} - v(F(\nu))^{\frac{1}{2}}\right|^2 \leq \left|v(F(\mu))^{\frac{1}{2}} - v(F(\nu))^{\frac{1}{2}}\right|\left|v(F(\mu))^{\frac{1}{2}} + v(F(\nu))^{\frac{1}{2}}\right| \tag{73}$$

$$\leq |v(F(\mu)) - v(F(\nu))|\,, \tag{74}$$

which after applying Cauchy–Schwarz results in an upper bound of the form:

$$\|K_{\mu,U_z} - K_{\nu,U_z}\|_2^2\|K_{U_z}^{-1}(K_{U_z-S})K_{U_z}^{-1}\|_2\,. \tag{75}$$

Using that $\|K_{\mu,U_z} - K_{\nu,U_z}\|_2^2 \leq \frac{16\sigma^4\lambda}{l^2}$ we obtain that:

$$|v(F(\mu)) - v(F(\nu))| \leq \frac{16\sigma^4\lambda}{l^2}\|K_{U_z}^{-1}(K_{U_z} - S)K_{U_z}^{-1}\|_2\,. \tag{76}$$

Now taking into consideration both the upper bounds on the mean and variance components we arrive at the desired Lipschitz constant.

□

B.1.2 Lipschitz properties of affine operator

The subsequent proof makes use of the following useful lemma.

Lemma B.1 (P-norm lemma)**.** *We have the following relations between norms :* $\|x\|_2 \leq \|x\|_1$ *and* $\|x\|_1 \leq \sqrt{D}\|x\|_2$.

Proof of Proposition 4.3.2

Proof. Using the definition for Wasserstein-2 distances for the l.h.s of the inequality, we can re-express as follows:

$$W_2\left(f(\mu), f(\nu)\right) \leq \|m_1 A - m_2 A\|_2^2 + \|(\sigma_1^2 A^2)^{1/2} - (\sigma_2^2 A^2)^{1/2}\|_F^2\,, \tag{77}$$

which after rearranging terms and noticing that inside the Frobenius norm we have scalars, be-

comes:

$$W_2\left(f(\mu), f(\nu)\right) \leq \|(m_1 - m_2)A\|_2^2 + [\sigma_1^2 A^2)^{1/2} - (\sigma_2^2 A^2)^{1/2}]^2 \,. \tag{78}$$

We can now apply the Cauchy–Schwarz inequality for the part involving means and multiplying the right hand side with $\sqrt{C}$, which represents the number of channels, we get:

$$\begin{aligned} \|(m_1 - m_2)A\|_2^2 + [\sigma_1^2 A^2)^{1/2} - (\sigma_2^2 A^2)^{1/2}]^2 \leq \\ \|m_1 - m_2\|_2^2 \sqrt{C}\|A\|_2^2 + \sqrt{C}[\sigma_1^2 A^2)^{1/2} - (\sigma_2^2 A^2)^{1/2}]^2 \,. \end{aligned} \tag{79}$$

We can notice that the Lipschitz constant for the component involving mean terms is $\sqrt{C}\|A\|_2^2$. Hence, we try to prove that the same L is also available for the variance terms component. Hence, we can affirm that:

$$L = \sqrt{C}\|A\|_2^2 \leftrightarrow \sqrt{C}[\sigma_1^2 A^2)^{1/2} - (\sigma_2^2 A^2)^{1/2}]^2 \leq [\sigma_1 - \sigma_2]^2 \sqrt{C}\|A\|_2^2 \,. \tag{80}$$

Using Cauchy–Schwarz we have $\sqrt{C}[\sigma_1 A - \sigma_2 A]^2 \leq [\sigma_1 - \sigma_2]^2 \sqrt{C}\|A\|_2^2$. Hence the aforementioned if and only if statement will hold if we prove that

$$\sqrt{C}\left[(\sigma_1^2 A^2)^{\frac{1}{2}} - (\sigma_2^2 A^2)^{\frac{1}{2}}\right]^2 \leq \sqrt{C}\left[\sigma_1 A - \sigma_2 A\right]^2 \,, \tag{81}$$

which after expressing in terms of norms becomes:

$$\sqrt{C}\left[\|\sigma_1 A\|_2 - \|\sigma_2 A\|_2\right]^2 \leq \sqrt{C}\left[\|\sigma_1 A\|_1 - \|\sigma_2 A\|_1\right]^2 \,. \tag{82}$$

Expanding the square brackets gives:

$$\begin{aligned} \sqrt{C}\left[\|\sigma_1 A\|_2^2 + \|\sigma_2 A\|_2^2 - 2\|\sigma_1 A\|_2 \|\sigma_2 A\|_2 \leq \right. \\ \left. \sqrt{C}\left[\|\sigma_1 A\|_1^2 + \|\sigma_2 A\|_1^2 - 2\|\sigma_1 A\|_1 \|\sigma_2 A\|_1\right] \right. \,. \end{aligned} \tag{83}$$

This inequality holds by applying the p-norm lemma, thereby the if and only if statement is satisfied. Consequently, the Lipschitz constant is $\sqrt{C}\|A\|_2^2$.

□

B.1.3 Feature-collapse *in DistGP Layers*

Proof of Proposition 4.3.3

Proof. We are interested in determining the specific scenarios in which the function space col-

lapses to constant values. Hence we explicitly write $\mathbb{E}\left[\|u_l(x) - u_l(x^*)\|_2^2 \mid u_{l-1}\right]$ as:

$$= \sum_{j=1}^{D_l} \mathbb{E}\left[\|u_l^j(x) - u_l^j(x^*)\|_2^2 \mid u_{l-1}\right] \tag{84}$$

$$= \sum_{j=1}^{D_l} \mathbb{E}\left[\left(u_l^j(x)\right)^2 \mid u_{l-1}\right] - 2\mathbb{E}\left[u_l^j(x)u_l^j(x^*) \mid u_{l-1}\right] + \mathbb{E}\left[\left(u_l^j(x^*)\right)^2 \mid u_{l-1}\right] \tag{85}$$

$$= \sum_{j=1}^{D_l} \sigma_l^2 + m_l^2(x) - 2m_l(x)m_l(x^*) - 2k^{W_2}\left[\mu_l(x), \mu_l(x^*)\right] + \sigma_l^2 + m_l^2(x^*) \tag{86}$$

$$= \sum_{j=1}^{D_l} \left[m_l^j(x) - m_l^j(x^*)\right]^2 + 2\sigma_l^2 - 2\sigma_l^2 \exp - \left[\frac{\mid m_{l-1}^j(x) - m_{l-1}^j(x^*) \mid^2}{2l_l^2}\right], \tag{87}$$

where in the last equation we have ignored the variance part of the Wasserstein-2 kernel since the two variance terms are equal. We make use of the following inequality $1 - \exp -x \leq x$ for $x \geq 0$ and equality only in the case that $x = 0$, resulting in the following upper bound:

$$\mathbb{E}\left[\|u_l(x) - u_l(x^*)\|_2^2 | \{u_l\}_{l=1}^{l-1}\right] \leq \sum_{j=1}^{D_l} \left[m_l^j(x) - m_l^j(x^*)\right]^2 + \sigma_l^2 \frac{\mid m_{l-1}^j(x) - m_{l-1}^j(x^*) \mid^2}{l_l^2}. \tag{88}$$

We can now view the previously defined operator $\overline{m_l(x)W_l}$ as an inner product in vector space between a tiled version of $m_l(x)$ and a normalised version of W_l, more specifically:

$$\langle [m_l(x), \cdots, m_l(x)], \left[\frac{W_{l,1}}{m_l}, \cdots, \frac{W_{l,m_l D_{l-1}}}{m_l}\right] \rangle = \overline{m_l(x)W_l}, \tag{89}$$

where m_l is the number of dimensions caused by the affine embedding function Ψ_l in the l-th layer of the hierarchy.

Our current goal is to relate $m_l^j(\cdot)$ to $m_{l-1}^j(\cdot)$. We can now apply Cauchy-Schwarz to:

$$\left|m_l^j(x) - m_l^j(x^*)\right|^2 = \left|\overline{m_{l-1}(x)W_l} - \overline{m_{l-1}(x^*)W_l}\right|^2 \tag{90}$$

$$= \left|\left\langle [m_{l-1}(x) - m_{l-1}(x^*), \cdots, m_{l-1}(x) - m_{l-1}(x^*)], \left[\frac{W_{l,1}}{m_l}, \cdots, \frac{W_{l,m_l D_{l-1}}}{m_l}\right] \right\rangle\right|^2 \tag{91}$$

$$\leq D_{l-1}m_l \left[m_{l-1}(x) - m_{l-1}(x^*)\right]^2 * \langle \tilde{W}_l, \tilde{W}_l \rangle, \tag{92}$$

where in the last line we denoted $\tilde{W}_l = [\frac{W_{l,1}}{D_l}, \cdots, \frac{W_{l,D_{l-1}m_l}}{m_l}]$ to avoid cluttering.

We can now apply the previous result to equation (88):

$$\mathbb{E}\left[\|u_l(x) - u_l(x^*)\|_2^2| \mid u_{l-1}\right] \leq \sum_{j=1}^{D_l} m_l D_{l-1} \left(m_{l-1}(x) - m_{l-1}(x^*)\right)^2 * \langle \tilde{W}_l, \tilde{W}_l \rangle \tag{93}$$

$$+ \frac{\sigma_l^2}{l_l^2} \left| m_{l-1}^j(x) - m_{l-1}^j(x^*) \right|^2$$

$$\leq \sum_{j=1}^{D_l} \left[m_l D_{l-1} * \langle \tilde{W}_l, \tilde{W}_l \rangle + \frac{\sigma_l^2}{l_l^2} \right] \left| m_{l-1}^j(x) - m_{l-1}^j(x^*) \right|^2 . \tag{94}$$

We can now recursively apply the previously derived Cauchy-Schwarz based inequality to obtain:

$$\mathbb{E}\left[|| \, u_l(x) - u_l(x^*) \, ||_2^2 \, | \{u_l\}_{l=1}^{l-1}\right] \leq \left[m_l D_{l-1} * \langle \tilde{W}_l, \tilde{W}_l \rangle + \frac{\sigma_l^2}{2l_l^2} \right] \prod_{l=1}^{l-1} D_l m_l D_{l-1} \langle \tilde{W}_l, \tilde{W}_l \rangle \tag{95}$$

$$\left[m_1(x) - m_1(x^*)\right]^2 .$$

By Markov's inequality, for any $\epsilon > 0$ we have that:

$$P\left(|| \, u_{l+1}(x) - u_{l+1}(x^*) \, ||_2 \geq \epsilon\right) \leq \frac{1}{\epsilon^2} \left[m_l D_{l-1} * \langle \tilde{W}_l, \tilde{W}_l \rangle + \frac{\sigma_l^2}{2l_l^2} \right] \prod_{l=1}^{l-1} D_l m_l D_{l-1} \langle \tilde{W}_l, \tilde{W}_l \rangle \tag{96}$$

$$\left[m_1(x) - m_1(x^*)\right]^2 .$$

Then, only in the case that $\left[m_l D_{l-1} * \langle \tilde{W}_l, \tilde{W}_l \rangle + \frac{\sigma_l^2}{2l_l^2} \right] \leq 1$ and $D_l m_l D_{l-1} \langle \tilde{W}_l, \tilde{W}_l \rangle \leq$ is satisfied for intermediate layers, we can apply the first Borel-Cantelli lemma to obtain:

$$P\left(\cap_{l=1}^{\infty} \cup_{m=l}^{\infty} || \, u_m(x) - u_m(x^*) \, ||_2 \geq \epsilon\right) = 0 . \tag{97}$$

Lastly, we can express the following:

$$P\left(|| \, u_n(x) - u_n(x^*) \, ||_2 \to 0\right) = P\left(\cap_{k=1}^{\infty} \cup_{l=1}^{\infty} \cap_{m=l}^{\infty} || \, u_m(x) - u_m(x^*) \, ||_2 \leq \frac{1}{k} \right) \tag{98}$$

$$= 1 - P\left(\cup_{k=1}^{\infty} \cap_{l=1}^{\infty} \cup_{m=l}^{\infty} || \, u_m(x) - u_m(x^*) \, ||_2 \geq \frac{1}{k} \right) \tag{99}$$

$$\geq 1 - \sum_{k=1}^{\infty} P\left(\cap_{l=1}^{\infty} \cup_{m=l}^{\infty} || \, u_m(x) - u_m(x^*) \, ||_2 \geq \frac{1}{k} \right) = 1 . \tag{100}$$

From which we obtain the proof of our proposition: $P\left(|| \, u_n(x) - u_n(x^*) \, ||_2 \to 0\right) = 1$. □

B.2 Additional results on out-of-distribution detection

B.2.1 Out-of-distribution detection in MRI scans with brain tumors

Observations. *We provide additional plots for the task investigated in section 4.5.3 for DistGP-Seg and OVA-DM as they were the only models to provide decent outlier detection capabilities. We refer the reader to Figures B.8 and B.9. From case study A, we can see that OVA-DM is over-segmenting across all FPR levels almost randomly from outside the tumor area, whereas DistGP-Seg is over-segmenting at $FPR = \{1.0, 5.0\}$ areas around the margins of the ventricles. For case study B, at $FPR = \{0.5, 1.0, 5.0\}$ OVA-DM seems to be under-segmenting the tumor in comparison to DistGP-Seg. The same observation can be made again for case study C. Lastly, for case study D DistGP-Seg seems to be under-segmenting for $FPR = \{0.1, 0.5\}$.*

B.2.2 Out-of-distribution detection in MRI scans with white matter hyperintensities

Motivation. *Different intensities of white matter hyperintensities (WMH) can be observed in a brain MRI (see Figure B.10 for illustration), especially in Multiple Sclerosis (MS) lesions. Age-related WMH are also present in subjects with cognitive disorders and seldom present in elderly people. The task of segmenting WMH is important so as to monitor disease progression. Manual delineation of WMH is time-consuming, suffering also from both intra and inter observer variability (Tran et al. 2022). The goal goal of this subsection is to see if DistGP-Seg can detect WMH as outliers given that it was trained to segment brain tissue on healthy patients with very low volume of WMH.*

Data and pre-processing. *Brain MRI scans from the 2015 Longitudinal Multiple Sclerosis Lesion Segmentation Challenge (Sweeney et al. 2013) which comprises of FLAIR, PD, T2-weighted, and T1-weighted volumes from a total of 110 MR imaging studies (11 longitudinal studies each of 10 subjects). All participants gave written consent and were scanned as part of an institutional review board approved natural history protocol. For the purposes of the task at hand, we only use the baseline FLAIR scans. All FLAIR images are pre-processed with skull-stripping, N4 bias correction, rigid registration to MNI152 space and histogram matching between UKBB and BraTS. Finally, we normalize intensities of each scan via linear scaling of its minimum and maximum intensities to the [-1,1] range.*

Remarks. *The task of detecting WMH is considerably more difficult than detecting tumors, the latter usually presenting itself as a large blob, whereas the former constitutes of multiple non-contiguous areas of varying shapes. From Figure B.11 we can notice that the large connected WMH regions are reliably detected as outliers, with smaller disconnected WMH regions being only in some cases outlined. Another issue is over-segmentation, as seen in case study C.*

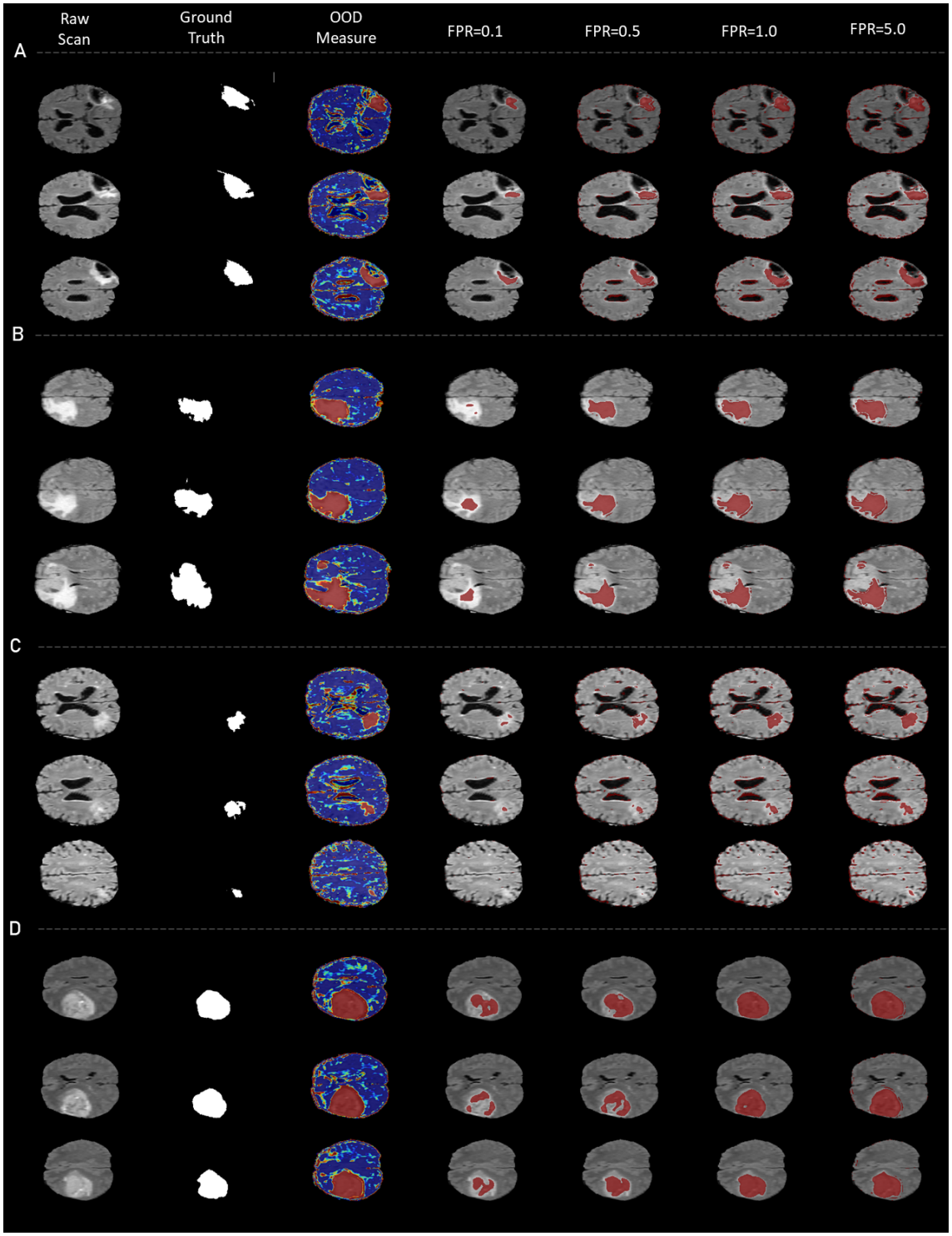


Figure B.8: Detailed segmentation output for DistGP-Seg on BraTS. Mean segmentation represents the hard segmentation of brain tissues. OOD measure is the quantification of uncertainty for each model, using their own procedure. Higher values (towards the red spectrum) translate to the respective tissue being detected as an outlier.

B.2.3 Out-of-distribution detection on Retina scans

Motivation. *Diabetic retinopathy (DR) is an eye condition that can cause vision loss and blindness in people who have diabetes, affecting the blood vessels in the retina. Hence, retinal screening*

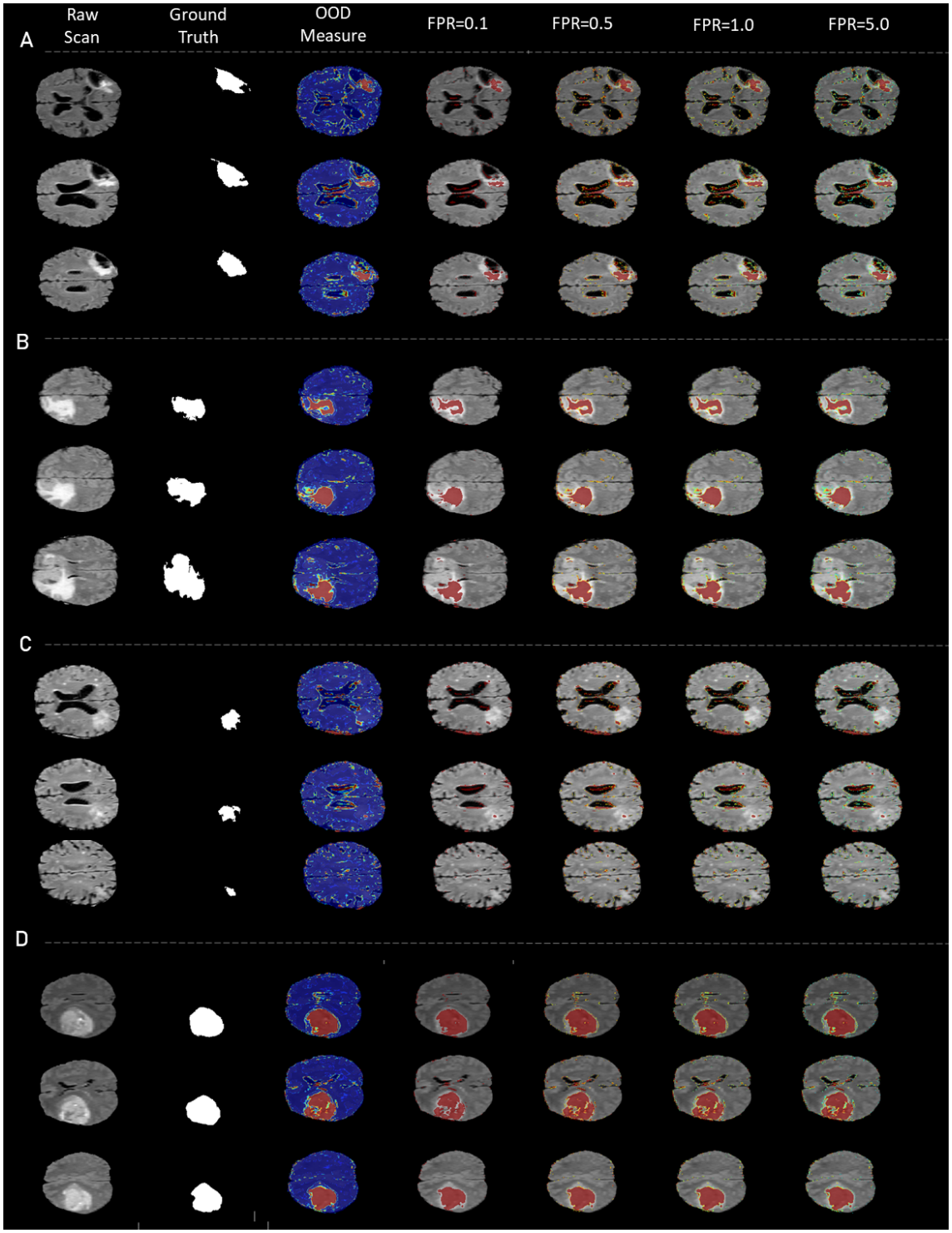


Figure B.9: Detailed segmentation output for OVA-DM on BraTS. Mean segmentation represents the hard segmentation of brain tissues. OOD measure is the quantification of uncertainty for each model, using their own procedure. Higher values (towards the red spectrum) translate to the respective tissue being detected as an outlier.

contributes to early detection of diabetic retinopathy and timely treatment. Mild non-proliferative DR (NPDR) is the early stage of DR, which is characterized by the presence of microaneurysms. Soft and hard exudates appear in advanced stages of NPDR (Alghadyan 2011). We provide an illustration

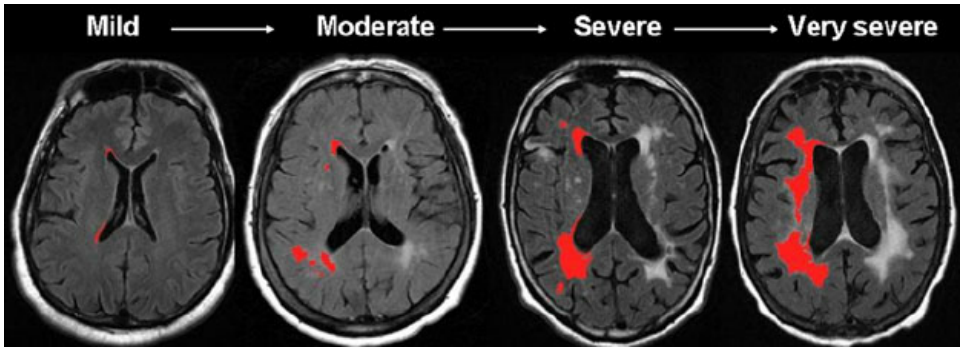


Figure B.10: Intensity scale of WMH lesions. WMH are areas with high signal intensities visible on T2-weighted fluid attenuated inversion recovery (T2-FLAIR) MRI sequences, being common findings in multiple sclerosis patients and elderly people.

of these retinal pathologies in Figure B.12. The goal of this subsection is to see if DistGP-Seg can detect these domain shifts in the case it was trained on healthy retinas.

Data and pre-processing. ***DRIVE:*** *The Digital Retinal Images for Vessel Extraction (DRIVE) dataset (Staal et al. 2004) is a dataset for retinal vessel segmentation. It consists of a total of JPEG 40 color fundus images; including 7 abnormal pathology cases. The images were obtained from a DR screening program in the Netherlands. The images were acquired using Canon CR5 non-mydriatic 3CCD camera with FOV equals to 45 degrees. Each image resolution is 584*565 pixels with eight bits per color channel (3 channels).*

The set of 40 images was equally divided into 20 images for the training set and 20 images for the testing set. Inside both sets, for each image, there is circular field of view (FOV) mask of diameter that is approximately 540 pixels. Inside training set, for each image, one manual segmentation by an ophthalmological expert has been applied. Inside testing set, for each image, two manual segmentations have been applied by two different observers, where the first observer segmentation is accepted as the ground-truth for performance evaluation.

STARE: *STructured Analysis of the Retina (STARE) database (Hoover et al. 2000) was created by scanning and digitizing the retinal image photographs. Hence, the image quality of this database is less than the other public databases. The STARE dataset comprises 97 images (59 AMD and 38 normal) taken using a fundus camera (TOPCON TRV-50; Topcon Corp., Tokyo, Japan) at a 35° field and with a resolution of* 605×700 *pixels. Its retina scans are from subjects suffering from the following retina pathologies: Hollenhorst Emboli, Branch Retinal Artery Occlusion, Cilio-Retinal Artery Occlusion, Branch Retinal Vein Occlusion, Central Retinal Vein Occlusion, Hemi-Central Retinal Vein Occlusion, Background Diabetic Retinopathy, Proliferative Diabetic Retinopathy, Arteriosclerotic Retinopathy, Hypertensive Retinopathy, Coat's, Macroaneurism, Choroidal Neovascularization.*

IDRID: *The Indian Diabetic Retinopathy Image Dataset (IDRID) dataset (Porwal et al. 2018), is a publicly available retinal fundus image database consisting of 516 images categorised into two parts:*

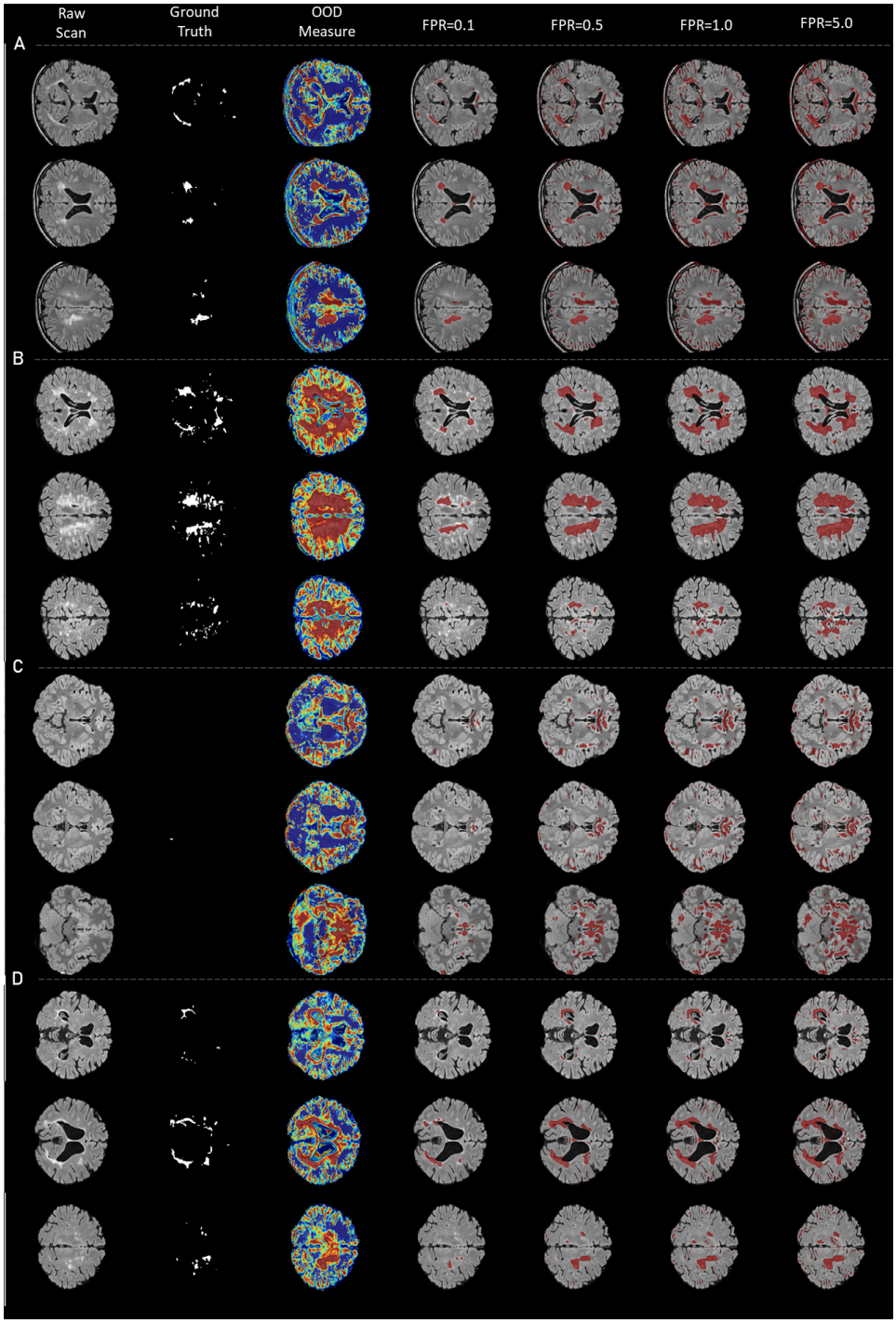


Figure B.11: Detailed segmentation output for DistGP-Seg on WMH dataset. Mean segmentation represents the hard segmentation of brain tissues. OOD measure is the quantification of uncertainty for each model, using their own procedure. Higher values (towards the red spectrum) translate to the respective tissue being detected as an outlier.

retina images with the signs of Diabetic Retinopathy and/or Diabetic Macular Edema; normal retinal images. Images were acquired using a Kowa VX-10a digital fundus camera with 50° field of view (FOV). The images have resolution of 4288×2848 *pixels and are stored in jpg file format. We have pre-processed these images to match the FOV and resolution of DRIVE.*

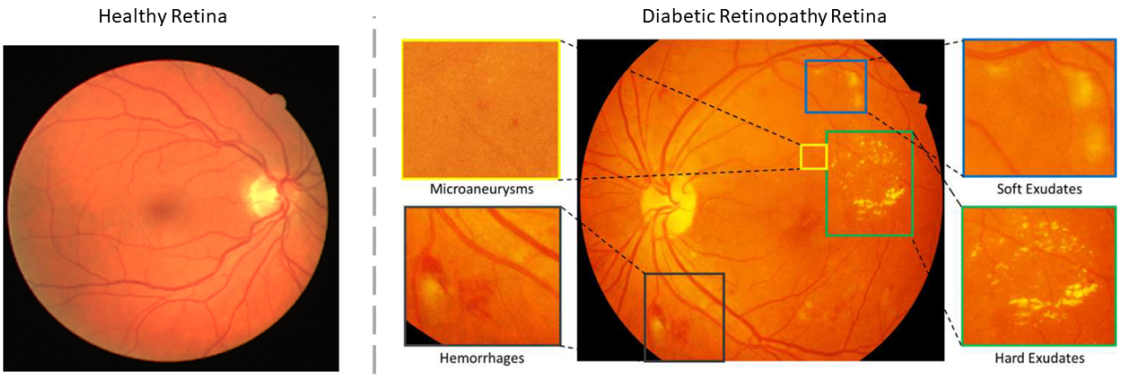


Figure B.12: Symptoms of Diabetic Retinopathy on retina scans.

Task. *We train a similar DistGP-Seg architecture (see section 4.5) adapted for 2D data on DRIVE (normative data) to segment blood vessels, subsequently at testing time we use STARE and IDRID (OOD data) to segment blood vessels in the presence of previously unseen pathologies on Retina scans. We would like to mention that we did not train the OOD baseline from literature (Amersfoort et al. 2020; Franchi et al. 2020; Liu et al. 2020a) as they only work on multi-class classification and not binary classification as is the case here.*

Blood vessel segmentation on normal Retina scans. *From Figure B.13 we can observe that DistGP-Seg manages to correctly segment the blood vessels, while both distributional and epistemic uncertainty are relatively low, which is to be expected as these testing examples represent in-distribution data.*

Out-of-distribution detection of retina pathologies. *From Figure B.14 we can observe that DistGP-Seg manages to correctly identity the vast majority of soft and hard exudates as outliers.*

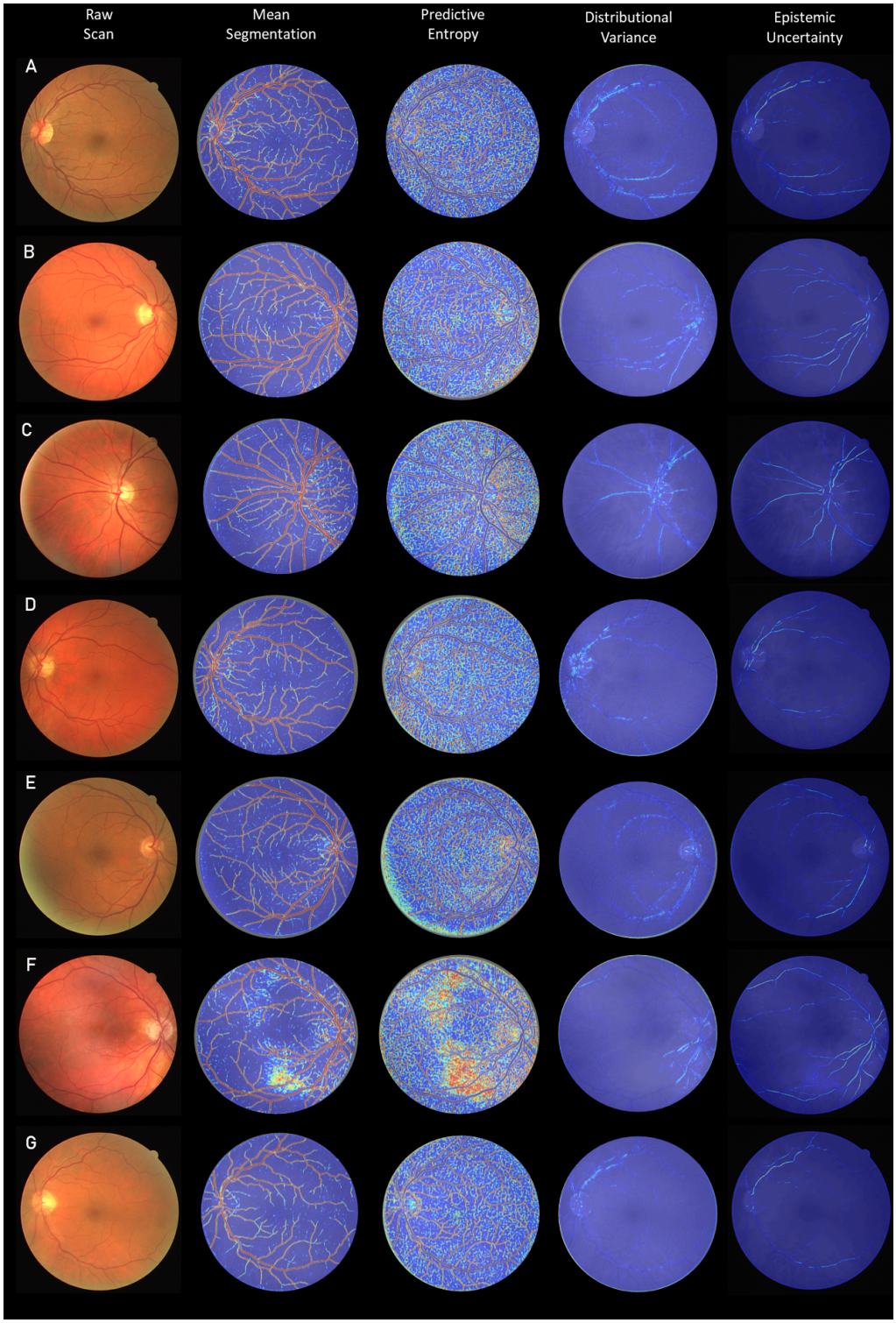


Figure B.13: Detailed segmentation output for DistGP-Seg on DRIVE dataset. Mean segmentation represents the hard segmentation of brain tissues. OOD measure is the quantification of uncertainty for each model, using their own procedure. Higher values (towards the red spectrum) translate to the respective tissue being detected as an outlier.

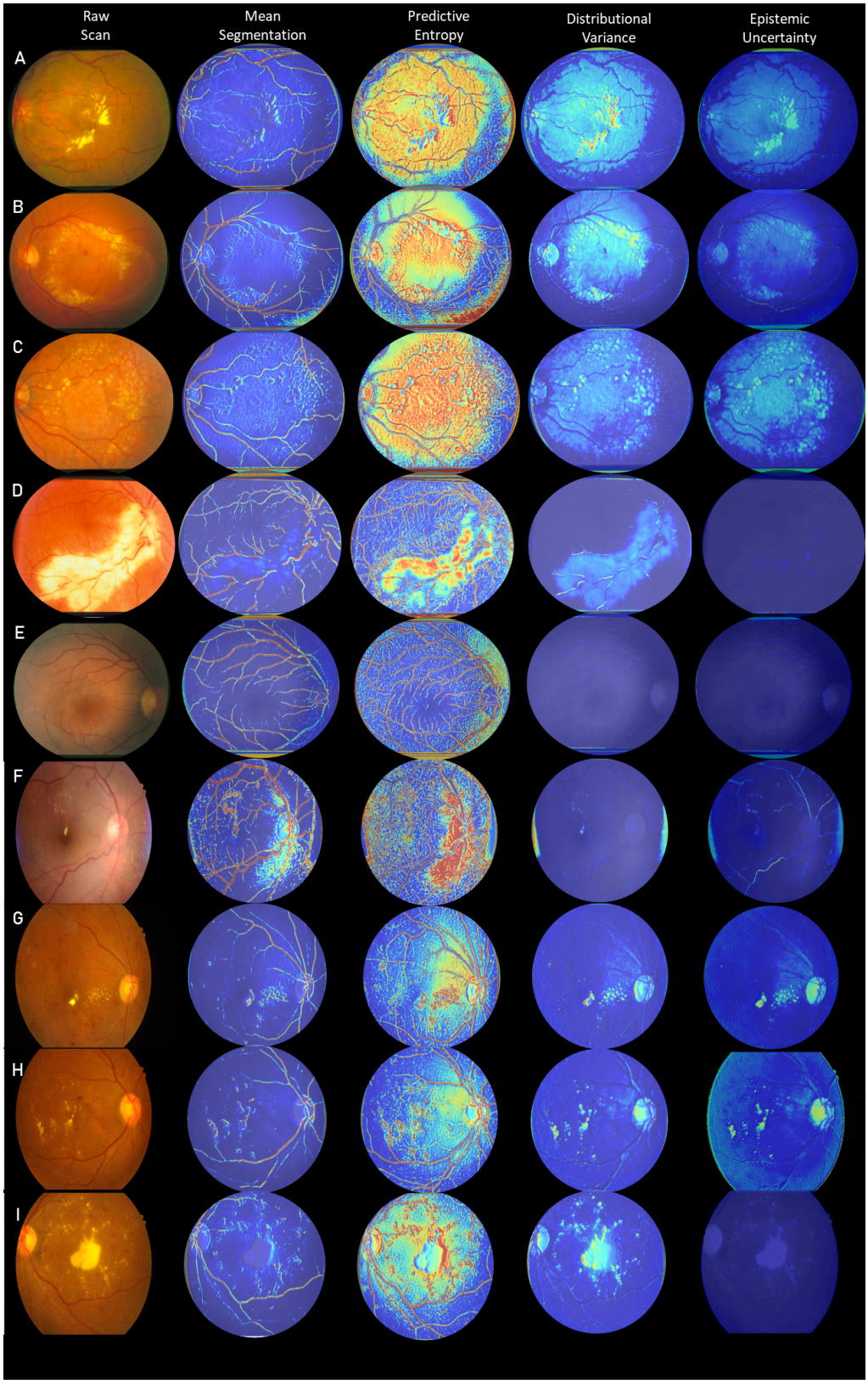


Figure B.14: Detailed segmentation output for DistGP-Seg on STARE/IDRID datasets. Mean segmentation represents the hard segmentation of brain tissues. OOD measure is the quantification of uncertainty for each model, using their own procedure. Higher values (towards the red spectrum) translate to the respective tissue being detected as an outlier. Case studies A-E originate from STARE, whereas the remainder from IDRID.

C Further details pertaining to inverse-free sparse Student-t Processes

In this section we provide all the necessary details regarding the generative process, ELBO, variational distribution and predictive equations for the IF-STP model briefly introduced in subsection 5.3.1. We remind ourselves that this model represented just a starting point for the subsequent derivation of IF-CSTP, which represents a computationally efficient version of IF-STP. Nevertheless, IF-STP is used in several experiments in section 5.6 mostly to compare its behaviour to that of IF-CSTP.

C.1 Alternative Variational Posteriors over Inverse-Wishart Processes

To perform inference over our hierarchical Bayesian model, we need a way of defining a valid variational mean covariance matrix for the posterior Inverse-Wishart Process over the space of possibly infinitely large covariance matrices: $q\left(\Sigma_{fu,fu}\right) = \mathcal{W}_{n+m}^{-1}\left(\nu_{f,q}+n, \nu_{f,q}\tilde{K}_{fu,fu}\right)$ *, where* $\tilde{K}_{fu,fu}$ *is a p.s.d. matrix defined as:*

$$\tilde{K}_{fu,fu} = \begin{pmatrix} \tilde{K}_{uu} & K_{uf} \\ K_{fu} & \tilde{K}_{ff} \end{pmatrix} = L_{\tilde{K}_{fu,fu}} L_{\tilde{K}_{fu,fu}}^{\top};\; L_{\tilde{K}_{fu,fu}} = \begin{pmatrix} L_{\tilde{K}_{uu}} & 0 \\ K_{fu} L_{\tilde{K}_{uu}}^{-\top} & L_{\tilde{K}_{ff\cdot u}} \end{pmatrix}, \tag{101}$$

where $L_{\tilde{K}_{uu}} L_{\tilde{K}_{uu}}^{\top} = \tilde{K}_{uu}$*,* $L_{\tilde{K}_{ff\cdot u}} L_{\tilde{K}_{ff\cdot u}}^{\top} = \tilde{K}_{ff\cdot u}$*,* $\tilde{K}_{ff} = \tilde{K}_{ff\cdot u} + K_{fu}\tilde{K}_{uu}^{-1}K_{uf}$ *and* $\nu_{f,q} = \nu_q + m + 1$*, with* $\nu_q \geq 0$*. Due to the block Cholesky construction,* $\tilde{K}_{fu,fu}$ *is guaranteed to be p.s.d. matrix, so that it satisfies the condition for defining mean covariance matrices for Inverse Wishart distributions.*

C.2 Variational lower bound of inverse free sparse Student-t Processes

In this subsection we are interested in obtaining a variational lower bound for the following posterior $q(F, U, \Sigma_{fu,fu}) = p\left(F|U, \Sigma_{fu,fu}\right) q\left(U\right) q(\Sigma_{fu,fu})$*, where:*

$$q\left(\Sigma_{fu,fu}\right) = \mathcal{W}_m\left(\nu_{f,q}+n,\; \frac{1}{\nu_{f,q}} \begin{bmatrix} L_{\tilde{K}_{uu}} L_{\tilde{K}_{uu}}^{\top} & K_{uf} \\ K_{fu} & \tilde{K}_{ff\cdot u} + K_{fu}\tilde{K}_{uu}^{-1}K_{uf} \end{bmatrix}\right) \tag{102}$$

$$q\left(U\right) = \mathcal{N}\left(m, L_S L_S^{\top}\right)\;. \tag{103}$$

We commence to derive the ELBO from scratch following the derivations initially introduced in Titsias (2009). We consider our training data $D = \{X_i, Y_i\}_{i=1,\cdots,n}$*. Our goal is to approximate the true*

posterior distribution $p(F, U, \Sigma_{fu,fu}^{-1}|D)$, which translates to minimizing the following:

$$= \int Q(F, U, \Sigma_{fu,fu}) \log \frac{Q(F, U, \Sigma_{fu,fu})}{p(F, U, \Sigma_{fu,fu}|D)} \mathrm{d}F \, \mathrm{d}U \, \mathrm{d}\Sigma_{fu,fu} \tag{104}$$

$$= \int Q(F, U, \Sigma_{fu,fu}) \left[\log \frac{Q(F, U, \Sigma_{fu,fu})}{p(F, U, \Sigma_{fu,fu}, D)} + \log P(D) \right] \mathrm{d}F \, \mathrm{d}U \, \mathrm{d}\Sigma_{fu,fu} \tag{105}$$

$$= \mathbb{E}_{Q(F,U,\Sigma_{fu,fu})} \left[\log \frac{Q(F, U, \Sigma_{fu,fu})}{p(D|F, U, \Sigma_{fu,fu}) p(F, U, \Sigma_{fu,fu}^{-1})} + \log P(D) \right] , \tag{106}$$

which after some rearrangements becomes:

$$\log p(D) - \left[\mathbb{E}_{q(F,U,\Sigma_{fu,fu})} \log p(D|F, U, \Sigma_{fu,fu}) - \mathrm{KL} \left[Q(F, U, \Sigma_{fu,fu}) \| p(F, U, \Sigma_{fu,fu}) \right] \right] . \tag{107}$$

We can now re-express this derivation as:

$$\log p(D) - \mathrm{KL} \left[q(F, U, \Sigma_{fu,fu} \| p(F, U, \Sigma_{fu,fu})) \right] = \tag{108}$$
$$\mathbb{E}_{q(F,U,\Sigma_{fu,fu})} \log p(D|F, U, \Sigma_{fu,fu}) - \mathrm{KL} \left[Q(F, U, \Sigma_{fu,fu}) \| p(F, U, \Sigma_{fu,fu}) \right] .$$

Hence, the standard variational approach Jordan et al. (1999) translates minimizing the KL divergence $\mathrm{KL} \left[q(F, U, \Sigma_{fu,fu} \| p(F, U, \Sigma_{fu,fu} \mid D) \right]$, which is intractable, to essentially maximizing a lower bound on the log marginal likelihood that is tractable, respectively the r.h.s. of the previous equation.

$$\mathcal{L}_{IF-STP} = \mathbb{E}_{Q(F,U,\Sigma_{fu,fu})} \log p(Y|F, U, \Sigma_{fu,fu}) - \mathbb{E}_{Q(F,U,\Sigma_{fu,fu})} \log \frac{q(F, U, \Sigma_{fu,fu})}{p(F, U, \Sigma_{fu,fu})} . \tag{109}$$

Our prior factorizes as $p(F, U, \Sigma_{fu,fu}) = p(F \mid U, \Sigma_{fu,fu}) p(U \mid \Sigma_{uu}^{-1}) p(\Sigma_{fu,fu})$. The reason for choosing the notation $p(U \mid \Sigma_{uu}^{-1})$ will become self evident later on in the computation of the KL-divergence term for U. In the KL term in equation (109) *we can separate the fraction terms that with respect to integrands that they exclusively depend on:*

$$\mathbb{E}_{p(F,U,\Sigma_{fu,fu})} \log \frac{\cancel{p(F|U, \Sigma_{fu,fu})}}{\cancel{p(F|U, \Sigma_{fu,fu})}} + \mathbb{E}_{q(\Sigma_{uu}^{-1})} \int q(U) \log \frac{q(U)}{p(U|\Sigma_{uu}^{-1})} \mathrm{d}U \tag{110}$$
$$+ \mathbb{E}_{q(\Sigma_{fu,fu})} \log \frac{q(\Sigma_{fu,fu})}{p(\Sigma_{fu,fu})}) ,$$

where the first term cancels out due to conditional prior matching. We now proceed to give detailed accounts of each term present in equation (109)*.*

Expectation over data fit term. *The expectation over the likelihood term can be expressed as:*

$$E_{q(\Sigma_{fu,fu})} \left[\int \log p(Y|F) q(F; \tilde{U}(\Sigma_{fu,fu}), \tilde{\Sigma}(\Sigma_{fu,fu})) \, \mathrm{d}F \right] , \tag{111}$$

where $\tilde{U}(\Sigma_{fu,fu}) = \Sigma_{fu}\Sigma_{uu}^{-1}m$ *and* $\tilde{\Sigma}(\Sigma_{fu,fu}) = \Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\left[\Sigma_{uu} - S\right]\Sigma_{uu}^{-1}\Sigma_{uf}$. *We can obtain samples for all matrix terms by using equations* (5.45) *and* (5.46) *adapted for the posterior, respectively:*

$$\Sigma_{ff\cdot u} \sim \mathcal{W}_n^{-1}\left(\nu_{f,p} + n,\ \nu_{f,p}\tilde{K}_{ff\cdot u}\right) \tag{112}$$

$$\Sigma_{uu}^{-1}\Sigma_{uf} \mid \Sigma_{ff\cdot u} \sim \mathcal{MN}\left(\tilde{K_{uu}^{-1}}K_{uf},\ \Sigma_{ff\cdot u} \otimes \frac{\tilde{K}_{uu}^{-1}}{\nu_{f,p}}\right) . \tag{113}$$

Marginalization of Inverse Wishart posterior. *In subsection C.1 we have defined a valid Inverse Wishart distribution over* $\Sigma_{fu,fu}$. *However, in the interest of performing S.V.I. with minibatches we are actually interested in Inverse Wishart distributions over* Σ_{f_1u,f_1u}, *whereby* f_1 *denotes a random sample from* f *at a given iteration in the training process and by* f_2 *we denote the remainder data points. Hence, under this scenario we can re-express* $\tilde{K}_{fu,fu}$ *as:*

$$\tilde{K}_{fu,fu} = \begin{pmatrix} \tilde{K}_{uu} & K_{uf_1} & K_{uf_2} \\ K_{f_1u} & v_1v_1^\top + K_{f_1u}\tilde{K}_{uu}^{-1}K_{uf_1} & v_1v_2^\top + K_{f_1u}\tilde{K}_{uu}^{-1}K_{uf_2} \\ K_{f_2u} & v_2v_1^\top + K_{f_2u}\tilde{K}_{uu}^{-1}K_{uf_1} & v_2v_2^\top + K_{f_2u}\tilde{K}_{uu}^{-1}K_{uf_2} \end{pmatrix} , \tag{114}$$

where $v = [v_1, v_2]$ *or more precisely, the original matrix can be decomposed into a concatenation along rows of the matrix* $v_1 \in M_{n_1\times n_1}$ *associated to* $f_1 \in M_{n_1\times 1}$ *and matrix* $v_2 \in M_{n_2\times n_2}$ *associated to* $f_2 \in M_{n_2\times 1}$ *repsectively.*

Due to marginalization properties of Inverse Wishart distributions we have:

$$\Sigma_{f_1u,f_1u} \sim \mathcal{W}^{-1}\left(\nu_q + n_1 + m + 1, \nu_q \begin{bmatrix} \tilde{K}_{uu} & K_{uf_1} \\ K_{f_1u} & v_1v_1^\top + K_{f_1u}\tilde{K}_{uu}^{-1}K_{uf_1} \end{bmatrix}\right) \tag{115}$$

$$\Sigma_{f_2f_2} \sim \mathcal{W}^{-1}\left(\nu_q + n_2 + 1, \nu_q\left[v_2v_2^\top + K_{f_2u}\tilde{K}_{uu}^{-1}K_{uf_2}\right]\right) . \tag{116}$$

Therefore, we can notice that if we marginalize out f_2 *we get a consistent mean covariance for* Σ_{f_1u,f_1u}. *The consistency under marginalization of a subset of* f *is given by the parametrization* $v_fv_f^\top + K_{fu}\tilde{K}_{uu}^{-1}K_{uf}$, *which can be easily permutated in its rows. Unlike Wishart distributions, the Inverse Wishart distribution is not consistent over degree of freedom under marginalization.*

Expectation in the SVI setting. *With these marginalization properties and consistency results, we can re-express expectations over individual terms as follows:*

$$\sum_{k=1}^{n} E_{q\left(\Sigma_{f_ku,f_ku}\right)}\left[\int \log p(Y_k|F_k)q(F_k;\tilde{U}(\Sigma_{f_ku,f_ku}),\tilde{\Sigma}(\Sigma_{f_ku,f_ku}))\,\mathrm{d}F_k\ ,\right] \tag{117}$$

where $\tilde{U}(\Sigma_{f_k u, f_k u}) = \Sigma_{f_k u}\Sigma_{uu}^{-1} m$ *and* $\tilde{\Sigma}(\Sigma_{f_k u, f_k u}) = \Sigma_{f_k f_k} - \Sigma_{f_k u}\Sigma_{uu}^{-1}\left[\Sigma_{uu} - S\right]\Sigma_{uu}^{-1}\Sigma_{u f_k}$ *define distributions over scalars. Hence, the formulas enunciated in equations* (112) *and* (113) *can now be rewritten as:*

$$\Sigma_{f_k f_k \cdot u} \sim Gamma^{-1}\left(\frac{\nu_{f,q}+1}{2},\ \frac{\nu_{f,q}}{2}\tilde{K_{f_k f_k \cdot u}}\right) \tag{118}$$

$$\Sigma_{uu}^{-1}\Sigma_{u f_k} \mid \Sigma_{f_k f_k \cdot u} \sim \mathcal{N}\left(\tilde{K}_{uu}^{-1} K_{u f_k},\ \frac{\Sigma_{f_k f_k \cdot u}}{\nu_{f,q}} \otimes \tilde{K}_{uu}^{-1}\right)\ , \tag{119}$$

where $Gamma^{-1}$ *represents in the inverse gamma distribution and both distributions are easy to sample from.*

KL-divergence between inducing points. *We now investigate* $\mathbb{E}_{q(\Sigma_{uu}^{-1})}\int q(U)\log\frac{q(U)}{p(U|\Sigma_{uu}^{-1})}\mathrm{d}U$*, which after sampling from* Σ_{uu}^{-1} *becomes:*

$$\mathrm{KL}\left[q(U)\|p(U \mid \Sigma_{uu}^{-1})\right] = \frac{1}{2}\left[-\log\left|\Sigma_{uu}^{-1}\right| - \log|S| - d + \mathrm{Tr}\left[\Sigma_{uu}^{-1}S\right] + m^\top\Sigma_{uu}^{-1}m\right]\ . \tag{120}$$

No matrix inversion is required for this divergence term. Moreover, Σ_{uu}^{-1} *is obtained via the Bartlett decomposition* $L_{\tilde{K}_{uu}^{-1}}AA^\top L_{\tilde{K}_{uu}^{-1}}^\top$*. Hence,* Σ_{uu}^{-1} *is a product of lower triangular matrices* $L_{\tilde{K}_{uu}^{-1}}A$*, with the determinant* $\log|\Sigma_{uu}^{-1}|$ *having an analytic formula.*

KL-divergence between covariance matrices. *We now focus on* $\mathrm{KL}\left[q(\Sigma_{fu,fu}\|p(\Sigma_{fu,fu}\right]$*, which has an analytic formula for Inverse Wishart distributions (Granström and Orguner 2012):*

$$\frac{\nu_q}{2}\log\left|\tilde{K}_{fu,fu}\right| - \frac{\nu_p}{2}\log|K_{fu,fu}| + \sum_{j=1}^{n+m}\left[\log\Gamma\left(\frac{\nu_p+1-j}{2}\right) - \log\Gamma\left(\frac{\nu_q+1-j}{2}\right)\right] + \tag{121}$$

$$\frac{\nu_p-\nu_q}{2}\left[\log\left|\tilde{K}_{fu,fu}\right| - \sum_{j=1}^{n+m}\psi_0\left(\frac{\nu_q+1-j}{2}\right)\right] + \mathrm{Tr}\left[-\frac{\nu_q}{2}\tilde{K}_{fu,fu}^{-1}\left(\tilde{K}_{fu,fu} - K_{fu,fu}\right)\right]\ .$$

Hence the problematic part is in computing the inverse of $\tilde{K}_{fu,fu}$*, which can be expressed as follows:*

$$\begin{pmatrix} \tilde{K}_{uu}^{-1} + \alpha_{fu}^\top\left(\tilde{K}_{ff\cdot u}\right)^{-1}\alpha_{fu} & -\alpha_{fu}^\top\left(\tilde{K}_{ff\cdot u}\right)^{-1} \\ -\left(\tilde{K}_{ff\cdot u}\right)^{-1}\alpha_{fu} & \left(\tilde{K}_{ff\cdot u}\right)^{-1} \end{pmatrix}\ , \tag{122}$$

where $\alpha_{fu} = K_{fu}\tilde{K}_{uu}^{-1}$*. Thereby, the initial problem boils down to computing inverse of lower triangular matrices* $L_{\tilde{K}_{uu}}$ *and* $L_{\tilde{K}_{ff\cdot u}}$*. Secondly, directly computing* $\log|K_{fu,fu}|$ *would incur an* $\mathcal{O}\left((N+M)^3\right)$ *computational cost. To bypass it, we use proposition 5.4.1.*

Remark C.1. *Translating Proposition 5.4.1 to our case, we have that* $log|K_{fu,fu}| \geq n + m +$

$\frac{1}{m}\sum_{i=1}^{m} -g_i^\top \tilde{K}_{fu,fu} g_i + log\left|\tilde{K}_{fu,fu}\right|$. *In the computation of* x_i *we precondition with* $\tilde{K}^{-1}_{fu,fu}$. *Hence, if* $\tilde{K}^{-1}_{fu,fu} \approx K^{-1}_{fu,fu}$ *the PCG routine will finish in one step.*

C.3 Obtaining predictions at testing time

We consider test points f^* *and hence we propose taking the following form over their posterior approximation of their Schur complement, namely* $\tilde{K}_{f^*f^*\cdot u} = K_{f^*f^*} + K_{f^*u}\tilde{K}^{-1}_{uu}K_{uu}\tilde{K}^{-1}_{uu}K_{uf^*} - 2K_{f^*u}\tilde{K}^{-1}_{uu}K_{uf^*}$, *which is p.s.d. matrix and equal to the exact Schur complement* $K_{f^*f^*\cdot u}$ *if and only if* $\tilde{K}^{-1}_{uu} = K^{-1}_{uu}$. *For an in-depth exploration into upper and lower bounds based on conjugate gradients we refer the reader to Davies (2015). In this case, we have the following probabilistic formulation:*

$$\Sigma_{f^*u,f^*u} \sim \mathcal{W}^{-1}\left(\nu_p + n^* + m + 1, (\nu_p + m + 1)\,\tilde{K}_{f^*u,f^*u}\right) , \tag{123}$$

where $\tilde{K}_{f^*u,f^*u}$ *is a p.s.d. matrix which can be defined as follows:*

$$\begin{pmatrix} \tilde{K}_{uu} & K_{uf^*} \\ K_{f^*u} & K_{f^*f^*} + K_{f^*u}\tilde{K}^{-1}_{uu}K_{uu}\tilde{K}^{-1}_{uu}K_{uf^*} - 2K_{f^*u}\tilde{K}^{-1}_{uu}K_{uf^*} + K_{f^*u}\tilde{K}^{-1}_{uu}K_{uf^*} \end{pmatrix} . \tag{124}$$

We can now write our full variational posterior over f^* *in a hierarchical manner:*

$$\Sigma_{f^*_k f^*_k\cdot u} \sim Gamma^{-1}\left(\frac{\nu_q + m + 2}{2},\ \frac{\nu_q + m + 1}{2}\tilde{K}_{f^*_k f^*_k\cdot u}\right) \tag{125}$$

$$\Sigma^{-1}_{uu}\Sigma_{uf^*_k} \mid \Sigma_{f^*_k f^*_k\cdot u} \sim \mathcal{N}\left(\tilde{K}^{-1}_{uu}K_{uf^*_k},\ \frac{\Sigma_{f^*_k f^*_k\cdot u}}{\nu_q + m + 1} \otimes \tilde{K}^{-1}_{uu}\right) \tag{126}$$

$$f^* \mid U, \Sigma_{f^*u,f^*u} \sim \mathcal{N}\left(\Sigma_{f^*u}\Sigma^{-1}_{uu}U, \Sigma_{f^*f^*\cdot u}\right) . \tag{127}$$

Taking samples from these distributions is a relatively easy task, with no computationally expensive operations.

D Natural Gradient update rules for various distributions

In this section we treat all required natural gradient update rule specific to the generative process introduced in section 5.3.

D.1 Natural Gradients for U

Exponential family representation of Multivariate Normal. *The Multivariate Normal distribution can be express as $q_\eta(\theta) = h(\theta)\exp\left[\langle\eta, \phi(\theta)\rangle - A(\eta)\right]$, where we have:*

- *Natural parameters* $\eta = \begin{pmatrix} \Sigma^{-1}\mu \\ -\frac{1}{2}\Sigma^{-1} \end{pmatrix}$
- *Base measure* $h(X) = (2\pi)^{-\frac{k}{2}}$
- *Sufficient statistic* $\phi(\theta) = \begin{pmatrix} X \\ XX^\top \end{pmatrix}$
- *log partition representation:* $A(\eta) = -\frac{1}{4}\eta_1^\top\eta_2^{-1}\eta_1 - \frac{1}{2}\log|-2\eta_2|$

Mean parameters of Multivariate Normal. *The mean parameters are given by:*

$$m = \begin{pmatrix} \mu \\ \mu\mu^\top + \Sigma \end{pmatrix} \tag{128}$$

Natural Gradients of Multivariate Normal with respect to data fit term. *We remind ourselves that under our posterior formulation the mean parameters can be expressed as:*

$$m = \begin{pmatrix} m \\ mm^\top + S \end{pmatrix} . \tag{129}$$

Therefore, we can express $\mu = m_1$ and $\Sigma = m_2 - \mu\mu^\top$. With this in mind, we can re-express gradients with respect to m as follows:

$$\frac{\partial L(\xi, y)}{\partial m_1} = \frac{\partial L(\xi, y)}{\partial \mu}\frac{\partial \mu}{\partial m_1} + \frac{\partial L(\xi, y)}{\partial L_S}\frac{\partial L_S}{\partial S}\frac{\partial S}{\partial m_1} \tag{130}$$

$$= \frac{\partial L(\xi, y)}{\partial \mu} + \frac{\partial L(\xi, y)}{\partial L_S}\left[L_S\Phi\left[L_S^{-1}E_{ij}L_S^{-\top}\right]\right][-2\mu] \tag{131}$$

$$\frac{\partial L(\xi, y)}{\partial m_2} = \frac{\partial L(\xi, y)}{\partial \mu}\frac{\partial \mu}{\partial m_2} + \frac{\partial L(\xi, y)}{\partial \Sigma}\frac{\partial L_S}{\partial S}\frac{\partial S}{\partial m_2} \tag{132}$$

$$= \frac{\partial L(\xi, y)}{\partial L_S}\left[L_S\Phi\left[L_S^{-1}E_{ij}L_S^{-\top}\right]\right] . \tag{133}$$

We remind ourselves that under our posterior formulation the natural parameters can be expressed as:

$$\eta = \begin{pmatrix} S^{-1}m \\ -\frac{1}{2}S^{-1} \end{pmatrix} . \tag{134}$$

Hence we can obtain the required derivatives as follows:

$$\frac{\partial m}{\partial \eta_1} = \frac{\partial S\eta_1}{\partial \eta_1} = S \tag{135}$$

$$\frac{\partial L_S}{\partial \eta_2} = \frac{\partial L_S}{\partial S}\frac{\partial S}{\partial \eta_2} = \frac{\partial L_S}{\partial S}\frac{\partial - \frac{\eta_2^{-1}}{2}}{\eta_2} \tag{136}$$

$$= \left[L_S \Phi \left[L_S^{-1} E_{ij} L_S^{-\top}\right]\right]\left[0.5\eta_2^{-1} E_{i,j} \eta_2^{-1}\right] \tag{137}$$

$$= \left[L_S \Phi \left[L_S^{-1} E_{ij} L_S^{-\top}\right]\right]\left[2SE_{i,j}S\right] \tag{138}$$

D.2 Natural Gradients for T

Exponential family representation of Wishart. *The Wishart distribution can be express as $q_\eta(\theta) = h(\theta)\exp\left[\langle \eta, \phi(\theta)\rangle - A(\eta)\right]$, where we have:*

- *Natural parameters:* $\eta = \begin{pmatrix} -\frac{1}{2}\phi^{-1} \\ \frac{\nu - m - 1}{2} \end{pmatrix} = \begin{pmatrix} -\frac{1}{2}\nu_f \tilde{K}_{uu} \\ \frac{\nu_f - m - 1}{2} \end{pmatrix}$
- *Base measure:* $h(X) = 1$
- *Sufficient statistic:* $\phi(\theta) = \begin{pmatrix} X \\ \log|X| \end{pmatrix}$
- *log partition representation:* $A(\eta) = -\left[\eta_2 + \frac{m+1}{2}\right]\log|-\eta_1| + \log\Gamma_m\left[\eta_2 + \frac{m+1}{2}\right]$

Mean parameters of Wishart distribution. *We commence to derive the mearn parameters of the Wishart distribution:*

$$m_1 = \frac{\partial A(\eta)}{\partial \eta_1} = \frac{\partial - \left[\eta_2 + \frac{m+1}{2}\right]\log|-\eta_1| + \log\Gamma_m\left[\eta_2 + \frac{m+1}{2}\right]}{\partial \eta_1} \tag{139}$$

$$= \frac{\partial - \left[\eta_2 + \frac{m+1}{2}\right]\log|-\eta_1|}{\partial \eta_1} = -\left[\eta_2 + \frac{m+1}{2}\right]\frac{\partial \log|-\eta_1|}{\partial - \eta_1}\frac{\partial - \eta_1}{\partial \eta_1} \tag{140}$$

$$= -\left[\eta_2 + \frac{m+1}{2}\right]\left[-\eta_1^{-\top}\right][-I] = -\left[\eta_2 + \frac{m+1}{2}\right]\eta_1^{-\top} \tag{141}$$

The second mean parameters is computed as follows:

$$m_2 = \frac{\partial A(\eta)}{\partial \eta_2} = -\log|-\eta_1| + \frac{\partial \log \Gamma_m\left[\eta_2 + \frac{m+1}{2}\right]}{\partial \eta_2} \tag{142}$$

$$= -\log|-\eta_1| + \frac{\partial \log \Gamma_m\left[\eta_2 + \frac{m+1}{2}\right]}{\partial \eta_2 + \frac{m+1}{2}} \frac{\partial \eta_2 + \frac{m+1}{2}}{\partial \eta_2} \tag{143}$$

$$= -\log|-\eta_1| + \sum_{i=1}^{m} \psi\left[\eta_2 + \frac{m+1}{2} + \frac{1-i}{2}\right] \tag{144}$$

$$= -\log|-\eta_1| + \sum_{i=1}^{m} \psi\left[\eta_2 + \frac{m+2-i}{2}\right], \tag{145}$$

where ψ represents the digamma function. We now gather the mean parameters:

$$m = \begin{pmatrix} -\left[\eta_2 + \frac{m+1}{2}\right] \eta_1^{-\top} \\ -\log|-\eta_1| + \sum_{i=1}^{m} \psi\left[\eta_2 + \frac{m+2-i}{2}\right] \end{pmatrix} \tag{146}$$

Natural Gradients of Wishart with respect to data fit term. *Under our posterior formulation the mean parameters become:*

$$m = \begin{pmatrix} \tilde{K}_{uu}^{-1} \\ -m\log\frac{v_f}{2} + \log|\tilde{K}_{uu}^{-1}| + \sum_{i=1}^{m} \psi\left[\frac{v_f+1-i}{2}\right] \end{pmatrix}. \tag{147}$$

To compute $\frac{\partial L(\tilde{\xi},y)}{\partial m_1}$ we make use of the chain rule in the following manner:

$$\frac{\partial L\left(\tilde{\xi}, y\right)}{\partial m_1} = \frac{\partial L\left(\tilde{\xi}, y\right)}{\partial \nu_f} \frac{\partial \nu_f}{\partial m_1} + \frac{\partial L\left(\tilde{\xi}, y\right)}{\partial L_{\tilde{K}_{uu}^{-1}}} \frac{\partial L_{\tilde{K}_{uu}^{-1}}}{\partial \tilde{K}_{uu}^{-1}} \tag{148}$$

$$= \frac{\partial L\left(\tilde{\xi}, y\right)}{\partial \nu_f} \frac{\partial \nu_f}{\partial m_1} + \frac{\partial L\left(\tilde{\xi}, y\right)}{\partial L_{\tilde{K}_{uu}^{-1}}} \left[L_{\tilde{K}_{uu}^{-1}} \Phi\left[L_{\tilde{K}_{uu}} E_{ij} L_{\tilde{K}_{uu}}^{\top}\right]\right]. \tag{149}$$

We now focus on computing $\frac{\partial \nu_f}{\partial m_1}$*:*

$$\frac{\partial \nu_f}{\partial m_1} = \frac{\partial \nu_f}{\partial - m \log \frac{\nu_f}{2} + \sum_{i=1}^{m} \psi\left(\frac{\nu_f+1-i}{2}\right)} \frac{\partial - m \log \frac{\nu_f}{2} + \sum_{i=1}^{m} \psi\left(\frac{\nu_f+1-i}{2}\right)}{\partial m_1} \tag{150}$$

$$= \left[\frac{\partial - m \log \frac{\nu_f}{2} + \sum_{i=1}^{m} \psi\left(\frac{\nu_f+1-i}{2}\right)}{\partial \nu_f}\right]^{-1} \frac{\partial m_2 - \log |m_1|}{\partial m_1} \tag{151}$$

$$= \left[-\frac{m}{\nu_f} + \sum_{i=1}^{m} \frac{\partial \psi\left(\frac{\nu_f+1-i}{2}\right)}{\partial \frac{\nu_f+1-i}{2}} \frac{\partial \frac{\nu_f+1-i}{2}}{\partial \nu_f}\right]^{-1} - m_1^{-1} \tag{152}$$

$$= \left[-\frac{m}{\nu_f} + \sum_{i=1}^{m} \frac{\psi_1\left(\frac{\nu_f+1-i}{2}\right)}{2}\right]^{-1} - \tilde{K}_{uu}\,, \tag{153}$$

where $\psi_1(\cdot)$ *is the trigamma function.*

To compute $\frac{\partial L(\tilde{\xi},y)}{\partial m_2}$ *we make use of the chain rule in the following manner:*

$$\frac{\partial L\left(\tilde{\xi}, y\right)}{\partial m_2} = \frac{\partial L\left(\tilde{\xi}, y\right)}{\partial \nu_f} \frac{\partial \nu_f}{\partial m_2} + \frac{\partial L\left(\tilde{\xi}, y\right)}{\partial L_{\tilde{K}_{uu}}} \frac{\partial L_{\tilde{K}_{uu}}}{\partial m_2} \tag{154}$$

$$= \frac{\partial L\left(\tilde{\xi}, y\right)}{\partial \nu_f} \frac{\partial \nu_f}{\partial - m \log \frac{\nu_f}{2} + \sum_{i=1}^{m} \psi\left(\frac{\nu_f+1-i}{2}\right)} \frac{\partial - m \log \frac{\nu_f}{2} + \sum_{i=1}^{m} \psi\left(\frac{\nu_f+1-i}{2}\right)}{\partial m_2} + 0 \tag{155}$$

$$= \frac{\partial L\left(\tilde{\xi}, y\right)}{\partial \nu_f} \left[-\frac{m}{\nu_f} + \sum_{i=1}^{m} \frac{\psi_1\left(\frac{\nu_f+1-i}{2}\right)}{2}\right]^{-1} \frac{\partial m_2 - \log |m_1|}{\partial m_2} \tag{156}$$

$$= \frac{\partial L\left(\tilde{\xi}, y\right)}{\partial \nu_f} \left[-\frac{m}{\nu_f} + \sum_{i=1}^{m} \frac{\psi_1\left(\frac{\nu_f+1-i}{2}\right)}{2}\right]^{-1}\,, \tag{157}$$

where $\psi_1(\cdot)$ *is the trigamma function.*