NEW ADVANCES IN BAYESIAN INFERENCE FOR GAUSSIAN PROCESS AND DEEP GAUSSIAN PROCESS MODELS

HAIBIN YU

(B.Eng., Beihang University)

A THESIS SUBMITTED FOR THE DEGREES OF

DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2020

New Advances in Bayesian Inference for Gaussian Process and Deep Gaussian Process Models

by

HAIBIN YU

Declaration

I hereby declare that this thesis is my original work
and it has been written by me in its entirety.

I have duly acknowledged all the sources of
information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

HAIBIN YU

January 14, 2020

Acknowledgements

The past five years in National University of Singapore (NUS) is an unforgettable and invaluable experience for me. Before I was enrolled in NUS, I barely knew anything about Singapore and had never heard of the research area of "machine learning" (not to mention Bayesian nonparametric methods and Gaussian processes, those words sounded totally alien to me). It seems surreal in retrospect that I would be doing research on Gaussian processes in the following five years, and would have completed this thesis about new advances in Bayesian inference for Gaussian process models.

First and foremost, I would like to express my greatest appreciation and gratitude to my supervisors, Professor Bryan Kian Hsiang Low and Professor Patrick Jaillet for their support, guidance, and advice throughout my PhD journey. Similar to life, a PhD journey is like a box of chocolate, and you never know what you are going to have. I still remember the exciting moment when receiving the admission email, the stressful hours to submit a paper right before the deadline at $4:00$ am, and the painful times to know a paper is rejected. There was a period that anxiety and depression got the best of me such that I almost withdrew the PhD project. However, Professor Bryan Kian Hsiang Low never gave up on me, his encouragement and support is of great value to me.

I would also like to express appreciation to Professor Harold Soh and Professor Tze Yun Leong for the constructive feedback which helps me in revising this thesis. Both of them are extremely enthusiastic and knowledgeable persons, it is my great honor to have them on my thesis committee.

I am also grateful for the willingness of sharing and insightful discussions with our group members, which keeps me informed of the latest research frontiers. In

particular, I would like to express my gratitude to Quoc Phong Nguyen, Yizhou Chen, and Zhongxiang Dai. Quoc Phong Nguyen is a smart person who has very strong mathematical backgrounds and the ability to see through chaos to discover the patterns, I enjoyed and learned a lot from the discussion with him. Yizhou Chen is two years junior to me, I am really impressed by his capability of designing novel solutions and plotting beautiful academical figures. Zhongxiang Dai is a mathematics-based person, and he is super enthusiastic about new knowledge and techniques in academia. I would like to thank him for his valuable proofreading and suggestions to my thesis.

I want to thank my parents: Zhong Yu and Hongguang Yin. Similar to most Chinese students among my generation, I am the only child of my family and I have a very close relationship with them. It always makes me feel guilty for only spending 2-3 weeks with them every year. I never know how to pay them back for their constant unconditional love, support, encouragement and care. I wish they are at least proud of me for what I have done so far. I would not be able to complete this journey without you.

Lastly, I would like to thank Dr. Miyang Luo for her love and support. We first met in Singapore during the tedious graduate English module in September, 2014. Both of us have a crush on each other from then. I still remember that unique day: 14th January, 2016 when our relationship starts. Interestingly, today is our four-year anniversary. I do enjoy the happy moments we spent together traveling around the world: Thailand, Japan, Indonesia, Malaysia, Taiwan, United States, Hungary, Czech, Canada. I do wish to hold your hand and walk along forever. Finally, I would like to ask sincerely, *Dr. Luo, will you marry me?*

Dedicated to my parents and Miyang, for their unconditional love and support

Contents

Abstract

Machine learning is the study of letting computers learn to perform a specific task in a data-driven manner. In particular, Bayesian machine learning has attracted enormous attention mainly due to their ability to provide uncertainty estimates following Bayesian inference. This thesis focuses on *Gaussian processes* (GPs), a rich class of Bayesian nonparametric models for performing Bayesian machine learning with formal measures of predictive uncertainty.

However, the applicability of GP in large datasets and in hierarchical composition of GPs is severely limited by computational issues and intractabilities. Therefore, it is crucial to develop accurate and efficient inference algorithms to address these challenges. To this end, this thesis aims at proposing a series of novel approximate Bayesian inference methods for a wide variety of GP models, which unifies the previous literatures, significantly extends them and hopefully lays the foundation for future inference methods.

To start with, this thesis presents a unifying perspective of existing inducing variables-based GP models, *sparse* GP (SGP) models and variational inference for SGP models (VSGP). Then, to further mitigate the issue of overfitting during optimization, we present a novel variational inference framework for deriving a family of Bayesian SGP regression models, referred to as *variational Bayesian SGP* (VBSGP) regression models.

Next, taking into account the fact that the expressiveness of GP and SGP depends heavily on the design of the kernel function, we further extend the expressive power of GP by introducing Deep GP (DGP), which is a hierarchical composition of GP models. Unfortunately, exact inference in DGP is intractable, which has motivated the recent development of deterministic and stochastic approximation methods. However, the deterministic approximation methods yield a biased posterior belief while the stochastic one is computationally costly. In this regard, we present the *implicit posterior variational inference* (IPVI) framework for DGPs that can ideally recover an unbiased posterior belief and still preserve time efficiency. Inspired by generative adversarial networks, our IPVI framework casts the DGP inference problem as a two-player game in which a Nash equilibrium, interestingly, coincides with an unbiased posterior belief.

Though IPVI will recover the unbiased posterior belief ideally, there is no guarantee that the best response dynamics algorithm, which is used to search for the optimal posterior belief, converges to the optimal solution in practice, which may result in suboptimal performance. Moreover, similar to the stochastic approximation methods, IPVI can only represent the posterior distribution through samples which lacks a crucial property, probability density, of a distribution. To this end, a novel and interesting variational inference framework for DGP models based on the notion of *Normalizing Flows* (NFs) is introduced. Interestingly, empirical experimental results reveal that the NF framework is more robust than IPVI in terms of training and outperforms IVPI in terms of predictive performance. This makes the NF framework a superior alternative to existing DGP methods.

We hope this thesis at least provides additional confidence and clarity for researchers who are devoting themselves to Bayesian nonparametric models, Gaussian process models in particular. Moreover, We also wish this thesis to offer inspirations for future works, and some thoughts that could be useful for future solutions.

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we firstly discuss the motivation and objective of this thesis and several concepts essential to this thesis such as: Bayesian machine learning, Bayesian nonparametrics and Gaussian processes are briefly introduced. Our aims are to emphasize the elegant properties of Bayesian machine learning and Gaussian processes which make them worthwhile to study. Then we summarize the contributions of this thesis. Finally, an outline of this thesis is provided.

1.1 Motivation

Machine learning is the study of statistical models for computers to perform a specific task effectively based on the observed data. For example, we provide the computers with images of dogs and would like to let the computers learn how to recognize dogs out of other images. Statistical models may contain parameters which are unknown to us, the aim is to (a) find the correct parameter settings which will improve the performance of the task, and (b) choose the "correct" statistical model which explains the data the best.

Generally, a statistical model is often viewed as *deterministic*. This means when

the model performs predictions (e.g. the model tries to tell whether a test image is a dog or not), it usually outputs a deterministic result (e.g. yes or no). However, for many machine learning tasks, it is often necessary to tell whether the model is certain about its predictions. The knowledge of whether a model is under-confident or overconfident (i.e. its uncertainty estimates are too small) serves as a good criteria to choose the correct model. More importantly, model uncertainty information can be used in critical systems where decisions may affect human life [Kendall and Gal, 2017]:

- In May 2016, the first fatality from an assisted driving system occurred, which is caused by the perception system confusing the white side of a trailer for bright sky.

- An image classification system erroneously identified two African Americans as gorillas, raising concerns of racial discrimination.

In the above-mentioned scenarios, relying on model uncertainty to adapt the decision making processes might be the key to preventing unintended behavior.

1.1.1 Bayesian Machine Learning

Fortunately, the critical issues mentioned above can be elegantly solved through Bayesian inference, which utilizes probability distributions to define the belief or uncertainty of unknown quantities in the model via the *prior*, and how they are related to the observed data via the *likelihood*. Bayesian inference turns the prior belief into the *posterior*, to represent the updated belief upon observing the data. The next section discusses Bayesian inference in more detail.

1.1.2 Bayesian Inference

Given a set of training inputs $\mathbf{x} = \{x_1, \ldots, x_N\}$ and their corresponding outputs $\mathbf{y} = \{y_1, \ldots, y_N\}$. Our objective is to train a model, parametrized by some unknown parameters θ, to capture the relationship between $\mathbf{x}$ and $\mathbf{y}$. such that when it comes to a new input x^*, our model can predict the correct output value y^*. In Bayesian statistics, the unknown parameters θ are considered as a random variable where we put some *prior* distribution over the space of parameters. Likewise, we further need to define the *likelihood* function $p(\mathbf{y}|\mathbf{x}, \theta)$—the probabilistic model by which the inputs $\mathbf{x}$ generate the outputs $\mathbf{y}$ given the parameters θ.

Given the dataset $\{\mathbf{x}, \mathbf{y}\}$, our goal is to improve our belief over the parameters θ following the *Bayes' rule*:

$$p(\theta|\mathbf{y}, \mathbf{x}) = \frac{p(\mathbf{y}|\mathbf{x}, \theta)p(\theta)}{p(\mathbf{y}|\mathbf{x})} \tag{1.1}$$

where $p(\theta|\mathbf{y}, \mathbf{x})$ denotes the *posterior* distribution and $p(\mathbf{y}|\mathbf{x})$ is called the *marginal likelihood* which is independent from θ. Given the posterior distribution we can predict an output distribution for a new input x^* by:

$$p(y^*|x^*) = \int p(y^*|x^*, \theta)p(\theta|\mathbf{y}, \mathbf{x}) \, \mathrm{d}\theta. \tag{1.2}$$

In order to ensure the flexibility and expressiveness of the models parameterized by θ, which is crucial to the success of Bayesian inference, different strategies have been proposed that can be categorized as either Bayesian parametric models or Bayesian nonparametric models, which will be discussed in the next section.

1.1.3 Bayesian Parametric Models and Bayesian Nonparametric Models

Bayesian parametric models assume a finite number of parameters, which makes the flexibility of the models constrained by the expressiveness of the number of parameters. In particular, the model parameters, which are inferred using the training data and applied to the test data for prediction, have a fixed size which is independent of the training set. This inflexibility of the model size might be restrictive since when the training size grows, the fixed-size parameters can become a limited information channel from the training data to the prediction on the test data. In contrast, Bayesian nonparametric models assume that the data distribution cannot be defined in terms of a finite set of parameters, and thus assume an infinite-dimensional parameter. One question naturally arises, which is how we can represent and manipulate such a *big* parameter. Fortunately, this parameter can be theoretically represented as a function. Moreover, the amount of information that this function can capture grows as the amount of data is increased. Generally, Bayesian nonparametric models are considered to be more flexible than Bayesian parametric models. In particular, Gaussian processes (GP), a rich class of Bayesian nonparametric models, is reviewed thoroughly in this thesis.

1.1.4 Gaussian Process

As we mentioned previously, a Gaussian process (GP) model is a rich class of Bayesian nonparametric models that can exploit correlation of the data for performing probabilistic machine learning by providing formal measures of predictive uncertainty.

Formally, a GP is a collection of random variables, any finite number of which follow a joint Gaussian distribution [Rasmussen and Williams, 2006]. A GP is completely specified by its mean function and covariance/kernel function which could be

written as:

$$\begin{aligned} f &\sim \mathcal{GP}(m_\theta(\mathbf{x}), k_\theta(\mathbf{x}, \mathbf{x}')) \\ m_\theta(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] \\ k_\theta(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m_\theta(\mathbf{x}))(f(\mathbf{x}') - m_\theta(\mathbf{x}'))]. \end{aligned} \tag{1.3}$$

where $\mathbf{x}$ and $\mathbf{x}'$ denote different input locations where the function f is evaluated, and θ denotes the set of hyperparameters of the mean and kernel functions. A common choice for the mean function is $m_\theta(\mathbf{x}) = 0$[1], since the prior knowledge and uncertainty about the mean function can be taken into account by adjusting the kernel function. After accounting for the mean function, the GP is fully specified by the form of the covariance/kernel function and the associated hyperparameters θ. The kernel determines how the model generalizes, or extrapolates to new data, and a variety of kernel functions have been proposed. For example, linear regression, splines and Kalman filters can all be considered specific realizations of GPs with particular kernels. A widely used kernel is the exponentiated quadratic or squared exponential (SE) kernel with automatic relevance determination (ARD)[2]:

$$k_\theta(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp(-0.5 \sum_{d=1}^{D} (x_d - x'_d)^2 / l_d). \tag{1.4}$$

where l_d is the lengthscale for the d-th input dimension, σ_f^2 is the kernel variance and the kernel hyperparameters $\theta = (\{l_d\}_{d=1}^D, \sigma_f)$.

Suppose we have a training dataset with D-dimensional inputs $\mathbf{X} \triangleq \{\mathbf{x}_n\}_{n=1}^N$ and their corresponding noisy observed scalar observations $\mathbf{y} \triangleq \{y_n\}_{n=1}^N$. A typical regression model assumes that each observation is the output from function f evaluated at input $\mathbf{x}_n$, which is corrupted by an additive, identically and independently distributed

[1]Note that throughout this thesis the mean function is set to be zero unless otherwise stated.
[2]Note that throughout this thesis we utilize the SE kernel unless otherwise stated.

(i.i.d) Gaussian noise ϵ:

$$\mathbf{y}_n = f(\mathbf{x}_n) + \epsilon.$$

where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ with noise variance σ_n^2. This induces the likelihood of the observations in the training set:

$$\begin{aligned} p(\mathbf{y}|\mathbf{f}) &= \mathcal{N}(\mathbf{f}, \sigma_n^2 \mathbf{I}) \\ p(\mathbf{f}) &= \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{XX}}) \end{aligned} \tag{1.5}$$

where $\mathbf{f} \triangleq \{f(\mathbf{x}_n)\}_{n=1}^N$ represents the latent function values and $\mathbf{K}_{\mathbf{XX}}$ denotes the covariance matrix with components $k_\theta(\mathbf{x}_n, \mathbf{x}_{n'})$ for $n, n' = 1, 2, \ldots, N$:

$$\mathbf{K}_{\mathbf{XX}} = \begin{pmatrix} k_\theta(\mathbf{x}_1, \mathbf{x}_1) & \ldots & k_\theta(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ k_\theta(\mathbf{x}_N, \mathbf{x}_1) & \ldots & k_\theta(\mathbf{x}_N, \mathbf{x}_N) \end{pmatrix}. \tag{1.6}$$

Following the **affine transformation** property in Appendix (A.2), the marginal probability of the observations $\mathbf{y}$ can be written as:

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{XX}} + \sigma_n^2 \mathbf{I}) \tag{1.7}$$

The task typically involves predicting the latent function value $f^\star = f(\mathbf{x}^\star)$ at unseen test input $\mathbf{x}^\star$. Since the joint distribution between the training observations $\mathbf{y}$ and latent function value $f^\star$ is a multivariate normal distribution, denoted as:

$$\begin{bmatrix} f^\star \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{\mathbf{x}^\star\mathbf{x}^\star} & \mathbf{K}_{\mathbf{x}^\star\mathbf{X}} \\ \mathbf{K}_{\mathbf{X}\mathbf{x}^\star} & \mathbf{K}_{\mathbf{XX}} + \sigma_n^2 \mathbf{I} \end{bmatrix} \right) \tag{1.8}$$

where $\mathbf{K}_{\mathbf{x}^\star\mathbf{x}^\star} = k_\theta(\mathbf{x}^\star, \mathbf{x}^\star)$ is the variance of the test function value, $\mathbf{K}_{\mathbf{x}^\star\mathbf{X}} \triangleq \mathbf{K}_{\mathbf{X}\mathbf{x}^\star}^\top$

and $\mathbf{K}_{\mathbf{x}^\star\mathbf{X}}$ denotes a vector with components $k_\theta(\mathbf{x}^\star, \mathbf{x}_n)$ for $n = 1, 2, \ldots, N$. Then the posterior distribution of $f^\star$ can be obtained according to the **conditioning** property of multivariate Gaussian distribution in Appendix (A.1):

$$p(f^\star|\mathbf{y}) = \mathcal{N}(\mathbf{K}_{\mathbf{x}^\star\mathbf{X}}(\mathbf{K}_{\mathbf{XX}} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}, \mathbf{K}_{\mathbf{x}^\star\mathbf{x}^\star} - \mathbf{K}_{\mathbf{x}^\star\mathbf{X}}(\mathbf{K}_{\mathbf{XX}} + \sigma_n^2\mathbf{I})^{-1}\mathbf{K}_{\mathbf{X}\mathbf{x}^\star}) \tag{1.9}$$

This suggests that the predictive distribution of the function value at an unseen test input is Gaussian-distributed with the posterior mean and variance given in (1.9) since (1.9) can be applied to any unseen input in the entire domain. In this regard, a GP can been seen as a prior over the function f. Conditioning this prior on the training data results in a posterior that 'fits' the data. Figure 1.1 demonstrates this procedure.

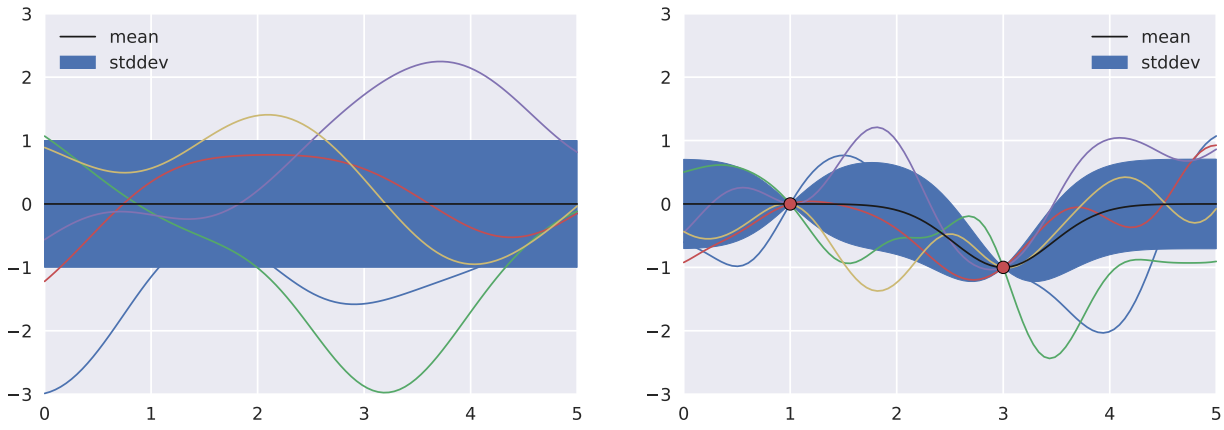


Figure 1.1: Left: functions drawn from a GP prior. Right: functions drawn from the GP posterior, after conditioning on two observations (red circles). The mean function is denoted by the black curve and two standard deviations are shown as the blue shading.

As shown in Figure 1.1, the entire inference procedure is based on a fixed set of kernel hyperparameters θ and noise variance σ_n^2. However, in practice, these parameters are not known beforehand and are complicated to set manually. Therefore, the

hyperparameters $\{\theta, \sigma_n^2\}$ are usually selected by maximizing the marginal likelihood $p(\mathbf{y})$, in order to improve the fitting of the GP to the observed data. The marginal likelihood $p(\mathbf{y})$ can be obtained by marginalizing out the variable $\mathbf{f}$ representing the latent function values variable $\mathbf{f}$:

$$\begin{aligned} p(\mathbf{y}) &= \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})\,\mathrm{d}\mathbf{f} \\ p(\mathbf{y}) &= (2\pi)^{-\frac{N}{2}}|\mathbf{K_{XX}} + \sigma_n^2\mathbf{I}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\mathbf{y}^\top(\mathbf{K_{XX}} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}\right) \end{aligned} \tag{1.10}$$

A closer inspection of expression of the marginal likelihood reveals two key properties: the determinant controls the model complexity by penalizing complex models, hence making the resulting hyperparameters robust against overfitting; the exponential term ensures the goodness-of-fit to the data.

1.2 Objective and Contributions

Despite the expressiveness and formal measurement of uncertainty, direct application of GP models is highly restrained by its computational and memory requirements. Specifically, computing the inversion of $\mathbf{K_{XX}} + \sigma_n^2\mathbf{I}$ shown in (1.9) and (1.10) costs $\mathcal{O}(N^3)$, and optimizing (1.10) requires computing this inversion operation multiple times. Therefore, the practical application of GP is usually limited to small datasets (e.g number of data points are less than 1000).

To this end, the first objective of this thesis is concerned with developing efficient approximation methods to enable the practical usage of GP to big datasets (e.g. million-size datasets). Moreover, we further extend the expressiveness of GP by introducing Deep GP (DGP), which is a hierarchical composition of GP models. Unfortunately, exact inference in DGP is intractable. In this regard, the other objective of this thesis is to propose accurate and effective approximation methods for DGP.

The main contributions of this thesis are summarized in the next three sections.

1.2.1 Variational Bayesian Sparse Gaussian Process Regression

We presents a novel variational inference framework for deriving a family of Bayesian sparse Gaussian process (SGP) regression models whose approximations are variationally optimal with respect to the GP regression model enriched with various corresponding correlation structures of the observation noises. Our variational Bayesian SGP (VBSGP) regression models jointly treat the distributions of both the inducing variables and hyperparameters as variational parameters, which enables the decomposability of the variational lower bound that in turn can be exploited for stochastic optimization. Such a stochastic optimization involves iteratively following the stochastic gradient of the variational lower bound to improve its estimates of the optimal variational distributions of the inducing variables and hyperparameters (and hence the predictive distribution) of our VBSGP regression models and is guaranteed to achieve asymptotic convergence to them. We show that the stochastic gradient is an unbiased estimator of the exact gradient and can be computed in constant time per iteration, hence achieving scalability to big data. Next, we empirically evaluate the performance of our proposed framework on two real-world, massive datasets.

1.2.2 Implicit Posterior Variational Inference for Deep Gaussian Processes

Analogous to deep neural networks are richer models than generalized linear models, a multi-layer deep Gaussian process (DGP) model is a hierarchical composition of GP models with a greater expressive power. Exact DGP inference is intractable, which has motivated the recent development of deterministic and stochastic approximation methods. Unfortunately, the deterministic approximation methods yield a biased

posterior belief while the stochastic one is computationally costly. We presents an novel implicit posterior variational inference (IPVI) framework for DGPs that can ideally recover an unbiased posterior belief and still preserve time efficiency. Inspired by generative adversarial networks, our IPVI framework achieves this by casting the DGP inference problem as a two-player game in which a Nash equilibrium, interestingly, coincides with an unbiased posterior belief. This consequently inspires us to devise a best-response dynamics (BRD) algorithm to search for a Nash equilibrium (i.e., an unbiased posterior belief). Empirical evaluations show that IPVI outperforms the state-of-the-art approximation methods for DGPs.

1.2.3 Variational Inference for Deep Gaussian Processes with Normalizing Flows

Though IPVI will recover the unbiased posterior belief ideally, there is no guarantee that the BRD algorithm converges to a Nash equilibrium in practice, which may result in suboptimal performance. Moreover, similar to the stochastic approximation methods, IPVI can only represent the posterior distribution through samples and thus lacks representation of the probability density, which is a crucial property of a distribution. To this end, we propose a novel and interesting variational inference framework for DGP models based on the notion of *Normalizing Flow* (NF). Interestingly, experimental results reveal that NF is capable of recovering complex multi-modal distributions in terms of both probability density as well as samples. Empirical experimental results show that NF is more robust than IPVI in terms of training stability and outperforms IVPI in terms of predictive performance. This makes NF framework a superior alternative to existing DGP methods.

1.3 Outline

The remaining chapters of this thesis are organized as follows. Chapter 2 reviews the technical details of approximate inference and sparse Gaussian processes (SGP). Recent advances of variational inference (VI) for SGP are also included. Chapter 3 presents the Bayesian SGP regression model which aims at eliminating the issues of overfitting in point estimate (non-Bayesian) SGP regression model. Furthermore, to boost the expressiveness power of SGP, Chapter 4 proposes a novel IPVI framework for DGP model. Then, to further improve the training stability as well as to recover the representation of probability density, Chapter 5 investigates the NF framework for DGP. Finally, Chapter 6 summarizes this thesis and discusses future research directions for GP.

Chapter 2

Background and Related Works

In this chapter, we review the necessary background to understand the three novel methods proposed in this thesis. To start with, Section 2.1 discusses two popular deterministic approximation inference methods. Secondly, Section 2.2 reviews the sparse approximation methods for GP. Thirdly, Section 2.3 discussed about the variational SGP models. Then, to further reduce the computational overhead of SGP models, stochastic variants of SGP models have been proposed which is introduced in Section 2.4. Finally, a summary of this chapter is provided.

2.1 Approximate Inference

Despite the simplicity of applying the *Bayes' rule*, computing the posterior distribution is usually intractable due to the marginal likelihood $p(\mathbf{y}|\mathbf{x})$.

$$p(\mathbf{y}|\mathbf{x}) = \int p(\mathbf{y}|\mathbf{x}, \theta)p(\theta) \, \mathrm{d}\theta \tag{2.1}$$

which requires integration over θ. The integration can be done analytically if the prior is chosen to be conjugate to the likelihood. However, the likelihood is usually so

complex that the conjugate prior does not exist. This motivates the development of deterministic (e.g. variational inference and expectation propagation) and stochastic (e.g. Markov chain Monte Carlo) approximation methods. We mainly focus on the deterministic approximation methods. For a detailed review of stochastic approximation methods, we refer the reader to [Bishop, 2006].

2.1.1 Variational Inference

As we have discussed in Section 1.1.2, the true posterior distribution $p(\theta|\mathbf{x}, \mathbf{y})$ usually can not be computed analytically. Hence, we define an approximating *variational* distribution $q_\phi(\theta)$, parametrized by ϕ, to approximate the intractable posterior distribution $p(\theta|\mathbf{x}, \mathbf{y})$. The objective is to find the distribution within the family of variational distributions parameterized by ϕ that is closest to the true posterior distribution. To this end, the KL distance (i.e., a measurement of the distance between two distributions) between the variational distribution and the true posterior distribution:

$$\mathrm{KL}(q_\phi(\theta)||p(\theta|\mathbf{x}, \mathbf{y})) = \int q_\phi(\theta) \log \frac{q_\phi(\theta)}{p(\theta|\mathbf{x}, \mathbf{y})} \, \mathrm{d}\theta, \tag{2.2}$$

is minimized to recover the optimal $q_{\phi^*}(\theta)$. Subsequently the prediction is made using the resulting variational distribution $q_\phi(\theta)$:

$$p(y^*|x^*) = \int p(y^*|x^*, \theta) q_\phi(\theta) \, \mathrm{d}\theta \tag{2.3}$$

In practice, KL distance minimization is equivalently achieved by maximizing the *evidence lower bound* (ELBO) w.r.t. the variational parameters ϕ:

$$\mathcal{L}_{ELBO}(\phi) = \int q_\phi(\theta) p(\mathbf{y}|\mathbf{x}, \theta) \, \mathrm{d}\theta - \mathrm{KL}(q_\phi(\theta)||p(\theta)). \tag{2.4}$$

The first term (the expected likelihood of the observed data) corresponds to the quality of data fit which encourages $q_\phi(\theta)$ to explain the data well, while the last term serves as a regularization which prevents $q_\phi(\theta)$ from deviating too much from the prior $p(\theta)$. The last term represents the "Occam razor" principle by penalizing complex distributions. The above-mentioned procedure is called *variational inference* (VI).

2.1.2 Expectation Propagation

In this section, we discuss an alternative form of deterministic approximation method, known as *expectation propagation* (EP) [Minka, 2001]. Similar to the VI method in Section 2.1.1, EP also tries to minimize the KL distance but in the reversed form:

$$\mathrm{KL}(p(\theta|\mathbf{x},\mathbf{y})||q_\phi(\theta)). \tag{2.5}$$

Without loss of generality, we assume $q_\phi(\theta)$ is a member of the *exponential family* and can be written in the form:

$$q_\phi(\theta) = h(\theta)g(\phi)\exp(\phi^\top \mathbf{u}(\theta)) \tag{2.6}$$

where ϕ is called the *natural parameters* for the distribution, $\mathbf{u}(\theta)$ is some function of θ and $g(\theta)$ is a coefficient to ensure that the distribution is normalized:

$$\int h(\theta)g(\phi)\exp(\phi^\top \mathbf{u}(\theta))\ \mathrm{d}\theta = 1. \tag{2.7}$$

Taking derivative of the above equation w.r.t. ϕ, we have:

$$\nabla g(\phi)\int h(\theta)\exp(\phi^\top \mathbf{u}(\theta))\ \mathrm{d}\theta + g(\phi)\int h(\theta)\exp(\phi^\top \mathbf{u}(\theta))\mathbf{u}(\theta)\ \mathrm{d}\theta = 0. \tag{2.8}$$

Rearranging, the terms lead to:

$$-\nabla \ln g(\phi) = \mathbb{E}_{q_\phi(\theta)}[\mathbf{u}(\theta)] \tag{2.9}$$

Interestingly, the KL distance for EP is also a function of θ:

$$\mathrm{KL}(p(\theta|\mathbf{x},\mathbf{y})||q_\phi(\theta)) = -\ln g(\phi) - \phi^\top \mathbb{E}_{p_\phi(\theta|\mathbf{x},\mathbf{y})}[\mathbf{u}(\theta)] + \text{const} \tag{2.10}$$

where the constant term is independent of ϕ. We can minimize the KL distance by setting the derivative w.r.t. ϕ to be zero:

$$-\nabla \ln g(\phi) = \mathbb{E}_{p_\phi(\theta|\mathbf{x},\mathbf{y})}[\mathbf{u}(\theta)]. \tag{2.11}$$

Therefore, we can derive:

$$\mathbb{E}_{q_\phi(\theta)}[\mathbf{u}(\theta)] = \mathbb{E}_{p_\phi(\theta|\mathbf{x},\mathbf{y})}[\mathbf{u}(\theta)]. \tag{2.12}$$

This suggests that the optimal solution simply corresponds to matching the expected *sufficient statistics* $\mathbf{u}(\theta)$. For instance, if $q_\phi(\theta)$ is assumed to be a Gaussian distribution, then

$$\mathbf{u}(\theta) = (\theta, \theta^2)^\top \tag{2.13}$$

Therefore, minimizing the KL distance corresponds to setting the mean and covariance of $q_\phi(\theta)$ to be equal to those of $p(\theta|\mathbf{x},\mathbf{y})$ respectively. This is referred to as *moment matching*.

2.1.3 Discussion

So far, we have discussed two popular deterministic approximation methods: VI and EP. Both of these methods depend on the KL distance to optimize the variational approximating distribution. However, there is a huge difference in terms of uncertainty estimates (e.g. estimates of the covariance of the true posterior distribution) between these to methods. VI tends to underestimate the covariance while EP tends to overestimate the covariance. Figure 2.1[1] demonstrates the different behavior of

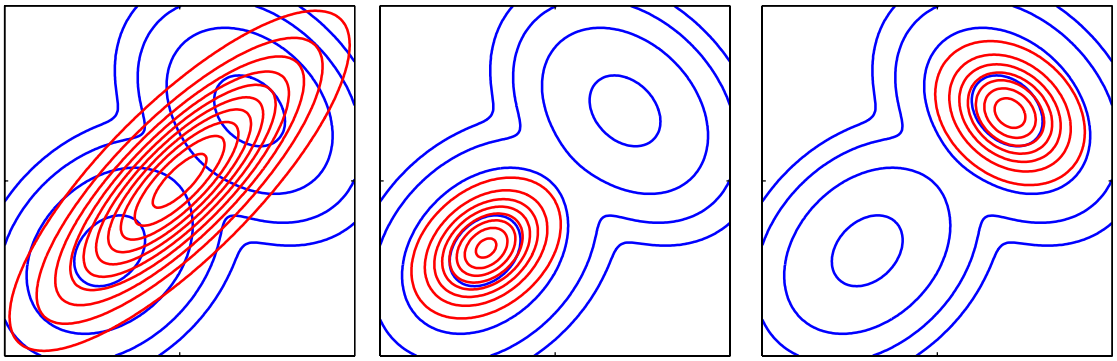


Figure 2.1: The blue contours demonstrate the bimodal posterior distribution $p(\theta|\mathbf{x},\mathbf{y})$, and the red contours corresponds to a Gaussian distribution which best approximates $p(\theta|\mathbf{x},\mathbf{y})$ by minimizing the KL distance. (a) corresponds to EP method. (b) and (c) denote different local optimal obtained using VI method.

these two methods if the posterior distribution is multi-modal and the approximating distribution is a unimodal.

The difference can be intuitively understood by taking a closer inspection of the KL distance:

$$\mathrm{KL}(q_\phi(\theta)||p(\theta|\mathbf{x},\mathbf{y})) = \int q_\phi(\theta) \log \frac{q_\phi(\theta)}{p(\theta|\mathbf{x},\mathbf{y})} \, \mathrm{d}\theta. \tag{2.14}$$

Note that there will be a large positive contribution to the KL distance from regions of θ in which $p(\theta|\mathbf{x},\mathbf{y})$ is near zero. Thus minimizing the KL distance in

[1]Figure is taken from [Bishop, 2006] for illustration.

VI methods encourages the approximating distribution $q_\phi(\theta)$ to avoid regions where $p(\theta|\mathbf{x}, \mathbf{y})$ is small. Conversely, for the EP method, $q_\phi(\theta)$ is optimized to cover all regions where $p(\theta|\mathbf{x}, \mathbf{y})$ is nonzero. Table 2.1 presents a summary of the different behavior between two methods.

	KL distance	covariance
VI	$\mathrm{KL}(q\|\|p)$	underestimate
EP	$\mathrm{KL}(p\|\|q)$	overestimate

Table 2.1: Summary of the different behavior between VI and EP.

2.2 Sparse Gaussian Processes

As mentioned in Section 1.1.4, a GP model, though highly expressive, incurs cubic time in the data size to compute the predictive distribution and to learn the hyperparameters via maximizing the marginal likelihood in (1.10). As a consequence, to learn the hyperparameters in reasonable time, only a very small subset of the data can be considered, which compromises the estimation accuracy, since the data subset is typically not representative of all the data in describing the underlying correlation structure due to its sparsity over the input space.

To improve the time efficiency of GP, a number of sparse Gaussian process (SGP) models exploiting low-rank covariance matrix approximations have been proposed, the majority of which [Csató and Opper, 2002; Seeger *et al.*, 2003; Snelson and Ghahramani, 2005; Quiñonero-Candela and Rasmussen, 2005; Snelson and Ghahramani, 2007] impose a common structural assumption of conditional independence (but of varying degrees) on the GP model. As a result, they incur linear time with respect to the data size. They key idea is to exploit the notion of *inducing variables* to approximate a low-rank representation of the covariance $\mathbf{K_{XX}}$. Specifically, we introduce a set $\mathbf{u} \triangleq \{u_m = f(\mathbf{z}_m)\}_{m=1}^M$ of inducing variables for some small set of

$\mathbf{Z} \triangleq \{\mathbf{z}_m\}_{m=1}^M$ of inducing inputs (i.e. $M \ll N$). Note that the inducing inputs $\mathbf{Z}$ are not necessarily a subset of the training inputs $\mathbf{X}$, they can be located anywhere within the input domain and thus can also be treated as trainable parameters and optimized. Equipped with the inducing variables, the joint distribution becomes:

$$\begin{aligned} p(\mathbf{y}, \mathbf{f}, \mathbf{u}) &= p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})p(\mathbf{u}) \\ p(\mathbf{u}) &= \mathcal{N}(\mathbf{0}, \mathbf{K_{ZZ}}) \end{aligned} \tag{2.15}$$

where $\mathbf{K_{ZZ}}$ denotes the covariance matrix with components $k_\theta(\mathbf{z}_m, \mathbf{z}_{m'})$ for $m, m' = 1, \ldots, M$.

The name inducing variables is motivated by the fact that $\mathbf{f}$ and $f^\star$ can only communicate through $\mathbf{u}$, and $\mathbf{u}$ therefore *induces* the dependencies between training and test data. Technically speaking, this implies that $\mathbf{f}$ and $f^\star$ are independent conditioned on $\mathbf{u}$, denoted as:

$$p(\mathbf{f}, f^\star|\mathbf{u}) = \mathcal{N}\left(\begin{bmatrix}\mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{u} \\ \mathbf{K_{x^\star Z}}\mathbf{K_{ZZ}^{-1}}\mathbf{u}\end{bmatrix}, \begin{bmatrix}\mathbf{K_{XX}} - \mathbf{Q_{XX}} & 0 \\ 0 & \mathbf{K_{x^\star x^\star}} - \mathbf{Q_{x^\star x^\star}}\end{bmatrix}\right) \tag{2.16}$$

where $\mathbf{Q_{XX}} \triangleq \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}}$ and $\mathbf{Q_{x^\star x^\star}} \triangleq \mathbf{K_{x^\star Z}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{Zx^\star}}$. $\mathbf{K_{XZ}} \triangleq \mathbf{K_{ZX}^\top}$, and $\mathbf{K_{ZX}}$ denote a matrix with components $k(\mathbf{z}_m, \mathbf{x}_n)$ for $m = 1, \ldots, M$ and $n = 1, \ldots, N$. $\mathbf{K_{x^\star Z}} \triangleq \mathbf{K_{Zx^\star}^\top}$, and $\mathbf{K_{Zx^\star}}$ denotes a vector with components $k(\mathbf{z}_m, \mathbf{x}^\star)$ for $m = 1, \ldots, M$.

As we shall detail in the following subsections, different sparse approximation methods correspond to different additional assumptions about the two conditionals $p(\mathbf{f}|\mathbf{u})$ and $p(f^\star|\mathbf{u})$, which is encompassed under a unifying view presented by [Quiñonero-Candela and Rasmussen, 2005] and [Snelson and Ghahramani, 2007]. It will be useful for future reference to specify here the exact expressions for the two

conditionals:

$$\begin{aligned} &\text{training conditional} : p(\mathbf{f}|\mathbf{u}) = \mathcal{N}(\mathbf{K_{XZ}K_{ZZ}^{-1}u}, \mathbf{K_{XX}} - \mathbf{Q_{XX}}) \\ &\text{test conditional} : p(f^\star|\mathbf{u}) = \mathcal{N}(\mathbf{K_{x^\star Z}K_{ZZ}^{-1}u}, \mathbf{K_{x^\star x^\star}} - \mathbf{Q_{x^\star x^\star}}) \end{aligned} \tag{2.17}$$

Note that the covariance matrices in (2.17) have the following interpretation: the prior covariance $\mathbf{K}$ minus a matrix $\mathbf{Q}$ which quantifies how much information $\mathbf{u}$ provides for $\mathbf{f}$ and $f^\star$.

2.2.1 Deterministic Training Conditional Approximation

The Deterministic Training Conditional (DTC) approximation was first proposed by [Seeger *et al.*, 2003]. In the original paper, Seeger *et al.* (2003) called this method 'Projected Latent Variables'. Equivalently, it corresponds to imposing a deterministic training conditional and using the exact test conditional modified from (2.17):

$$\begin{aligned} &\text{training conditional} : q_{\text{DTC}}(\mathbf{f}|\mathbf{u}) = \mathcal{N}(\mathbf{K_{XZ}K_{ZZ}^{-1}u}, \mathbf{0}) \\ &\text{test conditional} : q_{\text{DTC}}(f^\star|\mathbf{u}) = \mathcal{N}(\mathbf{K_{x^\star Z}K_{ZZ}^{-1}u}, \mathbf{K_{x^\star x^\star}} - \mathbf{Q_{x^\star x^\star}}) \end{aligned} \tag{2.18}$$

The joint prior conditioned on $\mathbf{u}$ implied by DTC is shown as:

$$q_{\text{DTC}}(\mathbf{f}, f^\star|\mathbf{u}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{K_{XZ}K_{ZZ}^{-1}u} \\ \mathbf{K_{x^\star Z}K_{ZZ}^{-1}u} \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{K_{x^\star x^\star}} - \mathbf{Q_{x^\star x^\star}} \end{bmatrix}\right). \tag{2.19}$$

Marginalizing out the inducing variables $\mathbf{u}$ according to the **affine transformation** property in Appendix (A.2) produces:

$$q_{\text{DTC}}(\mathbf{f}, f^\star) = \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{Q_{XX}} & \mathbf{Q_{Xx^\star}} \\ \mathbf{Q_{x^\star X}} & \mathbf{K_{x^\star x^\star}} \end{bmatrix}\right). \tag{2.20}$$

where $\mathbf{Q}_{\mathbf{x}^\star\mathbf{x}} = \mathbf{K}_{\mathbf{x}^\star\mathbf{Z}}\mathbf{K}_{\mathbf{ZZ}}^{-1}\mathbf{K}_{\mathbf{ZX}}$ and $\mathbf{Q}_{\mathbf{x}^\star\mathbf{x}} \triangleq \mathbf{Q}_{\mathbf{X}\mathbf{x}^\star}^\top$. Given $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{f}, \sigma_n^2\mathbf{I})$, the joint distribution between the training observations $\mathbf{y}$ and latent function value $f^\star$ can be written as:

$$q_{\mathrm{DTC}}(\mathbf{y}, f^\star) = \mathcal{N}\left(\begin{bmatrix}\mathbf{0}\\ 0\end{bmatrix}, \begin{bmatrix}\mathbf{Q}_{\mathbf{XX}} + \sigma_n^2\mathbf{I} & \mathbf{Q}_{\mathbf{X}\mathbf{x}^\star}\\ \mathbf{Q}_{\mathbf{x}^\star\mathbf{X}} & \mathbf{K}_{\mathbf{x}^\star\mathbf{x}^\star}\end{bmatrix}\right). \tag{2.21}$$

Following the **conditioning** property in Appendix (A.1), the predictive distribution can be derived as:

$$q_{\mathrm{DTC}}(f^\star|\mathbf{y}) = \mathcal{N}(\mathbf{Q}_{\mathbf{x}^\star\mathbf{X}}\left(\mathbf{Q}_{\mathbf{XX}} + \sigma_n^2\mathbf{I}\right)^{-1}\mathbf{y}, \mathbf{K}_{\mathbf{x}^\star\mathbf{x}^\star} - \mathbf{Q}_{\mathbf{x}^\star\mathbf{X}}\left(\mathbf{Q}_{\mathbf{XX}} + \sigma_n^2\mathbf{I}\right)^{-1}\mathbf{Q}_{\mathbf{X}\mathbf{x}^\star}) \tag{2.22}$$

If we apply the Woodbury and Matrix Determinant Lemma to the distribution above, a cheaper expression can be obtained:

$$q_{\mathrm{DTC}}(f^\star|\mathbf{y}) = \mathcal{N}(\sigma_n^{-2}\mathbf{K}_{\mathbf{x}^\star\mathbf{Z}}\mathbf{\Sigma}\mathbf{K}_{\mathbf{ZX}}\mathbf{y}, \mathbf{K}_{\mathbf{x}^\star\mathbf{x}^\star} - \mathbf{Q}_{\mathbf{x}^\star\mathbf{x}^\star} + \mathbf{K}_{\mathbf{x}^\star\mathbf{Z}}\mathbf{\Sigma}\mathbf{K}_{\mathbf{Z}\mathbf{x}^\star}) \tag{2.23}$$

where $\mathbf{Q}_{\mathbf{x}^\star\mathbf{x}^\star} = \mathbf{K}_{\mathbf{x}^\star\mathbf{Z}}\mathbf{K}_{\mathbf{ZZ}}^{-1}\mathbf{K}_{\mathbf{Z}\mathbf{x}^\star}$ and $\mathbf{\Sigma} = (\sigma_n^{-2}\mathbf{K}_{\mathbf{ZX}}\mathbf{K}_{\mathbf{XZ}} + \mathbf{K}_{\mathbf{ZZ}})^{-1}$. As can be seen, (2.23) only requires the inversion of an $M \times M$ matrix which considerably speeds up computations since $M \ll N$.

2.2.2 Fully Independent Training Conditional Approximation

Quiñonero-Candela and Rasmussen (2005) proposed another approximation called Fully Independent Training Conditional Approximation (FITC) based on the inducing conditionals:

$$\begin{aligned}&\text{training conditional}: q_{\mathrm{FITC}}(\mathbf{f}|\mathbf{u}) = \prod_{i=1}^{N} p(f_i|\mathbf{u}) = \mathcal{N}(\mathbf{K}_{\mathbf{XZ}}\mathbf{K}_{\mathbf{ZZ}}^{-1}\mathbf{u}, \mathrm{diag}[\mathbf{K}_{\mathbf{XX}} - \mathbf{Q}_{\mathbf{XX}}])\\ &\text{test conditional}: q_{\mathrm{FITC}}(f^\star|\mathbf{u}) = \mathcal{N}(\mathbf{K}_{\mathbf{x}^\star\mathbf{Z}}\mathbf{K}_{\mathbf{ZZ}}^{-1}\mathbf{u}, \mathbf{K}_{\mathbf{x}^\star\mathbf{x}^\star} - \mathbf{Q}_{\mathbf{x}^\star\mathbf{x}^\star})\end{aligned} \tag{2.24}$$

It can be observed that as opposed to DTC, FITC does not impose a deterministic relation between $\mathbf{f}$ and $\mathbf{u}$. Instead of ignoring the variance, FITC proposes an approximation to the training conditional distribution $p(\mathbf{f}|\mathbf{u})$ as a further independence assumption. Then the joint distribution between the training observations $\mathbf{y}$ and latent function value $f^\star$ for FITC can be derived as:

$$q_{\text{FITC}}(\mathbf{y}, f^\star) = \mathcal{N}\left(\begin{bmatrix}\mathbf{0}\\ 0\end{bmatrix}, \begin{bmatrix}\mathbf{Q_{XX}} + \text{diag}[\mathbf{K_{XX}} - \mathbf{Q_{XX}} + \sigma_n^2\mathbf{I}] & \mathbf{Q_{Xx^\star}}\\ \mathbf{Q_{x^\star X}} & \mathbf{K_{x^\star x^\star}}\end{bmatrix}\right). \tag{2.25}$$

Note that the sole difference between DTC and FITC is that in the top left corner of the implied prior covariance matrix, FITC replaces the approximate covariances of DTC by the exact ones on the diagonal. The predictive distribution is:

$$\begin{aligned} q_{\text{FITC}}(f^\star|\mathbf{y}) &= \mathcal{N}(\mathbf{Q_{x^\star X}}\left(\mathbf{Q_{XX}} + \mathbf{\Lambda}\right)^{-1}\mathbf{y}, \mathbf{K_{x^\star x^\star}} - \mathbf{Q_{x^\star X}}\left(\mathbf{Q_{XX}} + \mathbf{\Lambda}\right)^{-1}\mathbf{Q_{Xx^\star}})\\ &= \mathcal{N}(\mathbf{K_{x^\star Z}}\mathbf{\Sigma}\mathbf{K_{ZX}}\mathbf{\Lambda}^{-1}\mathbf{y}, \mathbf{K_{x^\star x^\star}} - \mathbf{Q_{x^\star x^\star}} + \mathbf{K_{x^\star Z}}\mathbf{\Sigma}\mathbf{K_{Zx^\star}}) \end{aligned} \tag{2.26}$$

where $\mathbf{\Sigma} = (\sigma_n^{-2}\mathbf{K_{ZX}}\mathbf{K_{XZ}} + \mathbf{K_{ZZ}})^{-1}$ and $\mathbf{\Lambda} = \text{diag}[\mathbf{K_{XX}} - \mathbf{Q_{XX}} + \sigma_n^2]$. The computational complexity is identical to that of DTC.

2.2.3 Partially Independent Training Conditional Approximation

Previously, FITC improves the DTC approximation by approximating the training conditional with an independent distribution, i.e. one with a diagonal covariance matrix. We can further relax the FITC assumption by partitioning the training data $\mathbf{X} = \{\mathbf{X}_{B_1}, \ldots, \mathbf{X}_{B_S}\}$. Then we can further improve the approximation (while maintaining the computational efficiency) by extending the training conditional to

have a block diagonal covariance:

$$\begin{aligned}
&\text{training conditional} : q_{\text{PITC}}(\mathbf{f}|\mathbf{u}) = \prod_{s=1}^{S} p(\mathbf{f}_{B_s}|\mathbf{u}) = \mathcal{N}(\mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{u}, \text{blockdiag}[\mathbf{K_{XX}} - \mathbf{Q_{XX}}]) \\
&\text{test conditional} : q_{\text{PITC}}(f^\star|\mathbf{u}) = \mathcal{N}(\mathbf{K_{x^\star Z}}\mathbf{K_{ZZ}^{-1}}\mathbf{u}, \mathbf{K_{x^\star x^\star}} - \mathbf{Q_{x^\star x^\star}})
\end{aligned} \tag{2.27}$$

where blockdiag$[A]$ is the block diagonal matrix for blocks $\{\mathbf{X}_{B_1}, \ldots, \mathbf{X}_{B_S}\}$. As it can be observed from the training conditional, FITC is a special case of PITC when the block size is 1. In fact, PITC [Snelson and Ghahramani, 2007] is a generalization of FITC. Then the joint distribution between the training observations $\mathbf{y}$ and the latent function value $f^\star$ for PITC is:

$$q_{\text{PITC}}(\mathbf{y}, f^\star) = \mathcal{N}\left(\begin{bmatrix}\mathbf{0}\\ 0\end{bmatrix}, \begin{bmatrix}\mathbf{Q_{XX}} + \text{blockdiag}[\mathbf{K_{XX}} - \mathbf{Q_{XX}} + \sigma_n^2] & \mathbf{Q_{Xx^\star}} \\ \mathbf{Q_{x^\star X}} & \mathbf{K_{x^\star x^\star}}\end{bmatrix}\right). \tag{2.28}$$

The predictive distribution can be written as:

$$\begin{aligned}
q_{\text{PITC}}(f^\star|\mathbf{y}) &= \mathcal{N}(\mathbf{Q_{x^\star X}}\left(\mathbf{Q_{XX}} + \mathbf{\Lambda}\right)^{-1}\mathbf{y}, \mathbf{K_{x^\star x^\star}} - \mathbf{Q_{x^\star X}}\left(\mathbf{Q_{XX}} + \mathbf{\Lambda}\right)^{-1}\mathbf{Q_{Xx^\star}}) \\
&= \mathcal{N}(\mathbf{K_{x^\star Z}}\mathbf{\Sigma}\mathbf{K_{ZX}}\mathbf{\Lambda}^{-1}\mathbf{y}, \mathbf{K_{x^\star x^\star}} - \mathbf{Q_{x^\star x^\star}} + \mathbf{K_{x^\star Z}}\mathbf{\Sigma}\mathbf{K_{Zx^\star}})
\end{aligned} \tag{2.29}$$

where $\mathbf{\Sigma} = (\sigma_n^{-2}\mathbf{K_{ZX}}\mathbf{K_{XZ}} + \mathbf{K_{ZZ}})^{-1}$ and $\mathbf{\Lambda} = \text{blockdiag}[\mathbf{K_{XX}} - \mathbf{Q_{XX}} + \sigma_n^2]$. The computational complexity is identical to that of DTC and FITC.

2.2.4 Partially Independent Conditional Approximation

All of the SGPs discussed so far have basically two assumptions: the conditional independence of $\mathbf{f}$ and $f^\star$ on $\mathbf{u}$; simplifying assumption on training conditional $p(\mathbf{f}|\mathbf{u})$. To further relax these assumptions, Snelson and Ghahramani (2007) derived a new

approximation by blocking the joint training and test conditional, denoted as:

$$\begin{aligned}\text{training conditional}: q_{\text{PIC}}(\mathbf{f}_{\cancel{B_S}}|\mathbf{u}) &= \prod_{s=1}^{S-1} p(\mathbf{f}_{B_s}|\mathbf{u}) \\ &= \mathcal{N}(\mathbf{K}_{\mathbf{X}_{\cancel{B_S}}\mathbf{Z}}\mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1}\mathbf{u}, \text{blockdiag}[\mathbf{K}_{\mathbf{X}_{\cancel{B_S}}\mathbf{X}_{\cancel{B_S}}} - \mathbf{Q}_{\mathbf{X}_{\cancel{B_S}}\mathbf{X}_{\cancel{B_S}}}]) \\ \text{test conditional}: q_{\text{PIC}}(f^\star, \mathbf{f}_{B_S}|\mathbf{u}) &= \\ \mathcal{N}(&\begin{bmatrix}\mathbf{K}_{\mathbf{x}^\star\mathbf{Z}}\mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1}\mathbf{u} \\ \mathbf{K}_{\mathbf{X}_{B_S}\mathbf{Z}}\mathbf{K}_{\mathbf{Z}\mathbf{Z}}^{-1}\mathbf{u}\end{bmatrix}, \begin{bmatrix}\mathbf{K}_{\mathbf{x}^\star\mathbf{x}^\star} - \mathbf{Q}_{\mathbf{x}^\star\mathbf{x}^\star} & \mathbf{K}_{\mathbf{x}^\star\mathbf{X}_{B_S}} - \mathbf{Q}_{\mathbf{x}^\star\mathbf{X}_{B_S}} \\ \mathbf{K}_{\mathbf{X}_{B_S}\mathbf{x}^\star} - \mathbf{Q}_{\mathbf{X}_{B_S}\mathbf{x}^\star} & \mathbf{K}_{\mathbf{X}_{B_S}\mathbf{X}_{B_S}} - \mathbf{Q}_{\mathbf{X}_{B_S}\mathbf{X}_{B_S}}\end{bmatrix}) \end{aligned} \tag{2.30}$$

Similar to the data partition of PITC in Section 2.2.3, $\mathbf{X} = \{\mathbf{X}_{B_1}, \ldots, \mathbf{X}_{B_S}\}$; $\mathbf{X}_{\cancel{B_S}}$ denotes training inputs excluding $\mathbf{X}_{B_S}$. The test conditional in PIC is modeled as a joint conditional distribution for $[f^*, \mathbf{f}_{B_S}]$. Marginalizing out the inducing variables $\mathbf{u}$ according to the **affine transformation** property in Appendix (A.2) generates:

$$q_{\text{PIC}}(f^\star, \mathbf{f}) = \mathcal{N}\left(\begin{bmatrix}0 \\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{K}_{\mathbf{x}^\star\mathbf{x}^\star} & \begin{pmatrix}\mathbf{K}_{\mathbf{x}^\star\mathbf{X}_{B_S}} & \mathbf{Q}_{\mathbf{x}^\star\mathbf{X}_{\cancel{B_S}}}\end{pmatrix} \\ \begin{pmatrix}\mathbf{K}_{\mathbf{X}_{B_S}\mathbf{x}^\star} \\ \mathbf{Q}_{\mathbf{X}_{\cancel{B_S}}\mathbf{x}^\star}\end{pmatrix} & \mathbf{Q}_{\mathbf{X}\mathbf{X}} + \mathbf{\Lambda}\end{bmatrix}\right). \tag{2.31}$$

The predictive distribution can be written as:

$$q_{\text{PIC}}(f^\star|\mathbf{y}) = \mathcal{N}(\tilde{\mathbf{K}}_{\mathbf{x}^\star\mathbf{X}}\left(\mathbf{Q}_{\mathbf{X}\mathbf{X}} + \mathbf{\Lambda}\right)^{-1}\mathbf{y}, \mathbf{K}_{\mathbf{x}^\star\mathbf{x}^\star} - \tilde{\mathbf{K}}_{\mathbf{x}^\star\mathbf{X}}\left(\mathbf{Q}_{\mathbf{X}\mathbf{X}} + \mathbf{\Lambda}\right)^{-1}\tilde{\mathbf{K}}_{\mathbf{X}\mathbf{x}^\star}) \tag{2.32}$$

where $\tilde{\mathbf{K}}_{\mathbf{x}^\star\mathbf{X}} = (\mathbf{K}_{\mathbf{x}^\star\mathbf{X}_{B_S}}, \mathbf{Q}_{\mathbf{x}^\star\mathbf{X}_{\cancel{B_S}}})$ and $\tilde{\mathbf{K}}_{\mathbf{x}^\star\mathbf{X}} \triangleq \tilde{\mathbf{K}}_{\mathbf{X}\mathbf{x}^\star}^\top$. Similarly, the computational complexity is the same as DTC, FITC, and PITC.

2.3 Variational Sparse Gaussian Process

The aforementioned unified SGP models (DTC, FITC, PITC, and PIC) essentially modifies the GP prior. The chief concern with the unifying view pointed out by

[Titsias, 2009] is that it does not rigorously quantify the approximation quality of an SGP model. To address this concern, Titsias (2009) proposed a VI framework that involves minimizing the KL distance between the variational posterior distribution $q(\mathbf{f}, \mathbf{u})$ and the true posterior distribution $p(\mathbf{f}, \mathbf{u}|\mathbf{y})$ denoted as $\mathrm{KL}[q(\mathbf{f}, \mathbf{u})||p(\mathbf{f}, \mathbf{u}|\mathbf{y})]$. According to what we have explained in Section 2.1.1, this is equivalent to maximizing the ELBO. Specifically, we can write:

$$\begin{aligned}\mathrm{KL}(q(\mathbf{f}, \mathbf{u})||p(\mathbf{f}, \mathbf{u}|\mathbf{y})) &= \iint q(\mathbf{f}, \mathbf{u}) \log \frac{q(\mathbf{f}, \mathbf{u})p(\mathbf{y})}{p(\mathbf{y}, \mathbf{f}, \mathbf{u})} \, \mathrm{d}\mathbf{f}\mathrm{d}\mathbf{u} \\ &= \log p(\mathbf{y}) - \iint q(\mathbf{f}, \mathbf{u}) \log \frac{p(\mathbf{y}, \mathbf{f}, \mathbf{u})}{q(\mathbf{f}, \mathbf{u})} \, \mathrm{d}\mathbf{f}\mathrm{d}\mathbf{u} \\ &= \log p(\mathbf{y}) - \mathcal{L}(q)\end{aligned} \tag{2.33}$$

where $\mathcal{L}(q)$ represents the ELBO. As can be seen, the sum of the KL distance and the ELBO is the log-marginal likelihood $\log p(\mathbf{y})$, which is a constant with respect to $q(\mathbf{f}, \mathbf{u})$. The variational approach differs from the SGP approaches which directly seek a low-rank approximation through assumptions on $q(\mathbf{f}|\mathbf{u})$, and modify the GP prior. In the VI approach, the variational posterior approximation can also be represented as:

$$q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f}|\mathbf{u})q(\mathbf{u}) \tag{2.34}$$

where $q(\mathbf{u})$ is a free variational Gaussian distribution according to [Titsias, 2009]. The ELBO can then be simplified as:

$$\begin{aligned}\mathcal{L}(q) &= \iint p(\mathbf{f}|\mathbf{u})q(\mathbf{u}) \log \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})p(\mathbf{u})}{p(\mathbf{f}|\mathbf{u})q(\mathbf{u})} \, \mathrm{d}\mathbf{f}\mathrm{d}\mathbf{u} \\ &= \iint p(\mathbf{f}|\mathbf{u})q(\mathbf{u}) \log \frac{p(\mathbf{y}|\mathbf{f})\cancel{p(\mathbf{f}|\mathbf{u})}p(\mathbf{u})}{\cancel{p(\mathbf{f}|\mathbf{u})}q(\mathbf{u})} \, \mathrm{d}\mathbf{f}\mathrm{d}\mathbf{u} \\ &= \mathbb{E}_{q(\mathbf{f})}[\log p(\mathbf{y}|\mathbf{f})] - \mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})], \quad q(\mathbf{f}) = \int p(\mathbf{f}|\mathbf{u})q(\mathbf{u}) \, \mathrm{d}\mathbf{u}.\end{aligned} \tag{2.35}$$

It has been proved by [Gal and van der Wilk, 2014] that under the regression frame-

work, the optimal $q^*(\mathbf{u})$ is Gaussian, denoted as:

$$q^*(\mathbf{u}) = \mathcal{N}(\sigma_n^{-2}\mathbf{K_{ZZ}}\mathbf{\Sigma}\mathbf{K_{ZX}}\mathbf{y}, \mathbf{K_{ZZ}}\mathbf{\Sigma}\mathbf{K_{ZZ}}) \tag{2.36}$$

where $\mathbf{\Sigma}$ is the same as that in (2.23). The ELBO can then be optimized analytically:

$$\mathcal{L}(q) = \log \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{Q_{XX}} + \sigma_n^2\mathbf{I}) - \frac{1}{2\sigma_n^2}\mathrm{Tr}(\mathbf{K_{XX}} - \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}}). \tag{2.37}$$

The variational SGP predictive distribution can also be computed in closed form by marginalizing out $\mathbf{u}$:

$$p(f^\star|\mathbf{y}) = \int p(f^\star|\mathbf{y})p(\mathbf{u}|\mathbf{y})\ \mathrm{d}\mathbf{u} \approx \int p(f^\star|\mathbf{y})q^*(\mathbf{u})\ \mathrm{d}\mathbf{u} \tag{2.38}$$

Interestingly, following the **affine transformation** property in Appendix (A.2), the predictive distribution is exactly the same as that of DTC discussed in Section 2.2.1. Therefore, in terms of predictive capability, variational SGP is equivalent to DTC. However, the explanation of the variational method is very different from that of the DTC approximation due to the extra regularization trace term: $-\frac{1}{2\sigma_n^2}\mathrm{Tr}(\mathbf{K_{XX}} - \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}})$ as shown in (2.37).

A closer inspection of (2.37) reveals that the ELBO is the sum of DTC log likelihood and the regularization trace term $-\frac{1}{2\sigma_n^2}\mathrm{Tr}(\mathbf{K_{XX}} - \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}})$. Therefore, the ELBO attempts to maximize the log likelihood, while in the meantime minimizing the trace $\mathrm{Tr}(\mathbf{K_{XX}} - \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}})$. Titsias (2009) pointed out the trace $\mathrm{Tr}(\mathbf{K_{XX}} - \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}})$ represents the total variance of the training conditional $p(\mathbf{f}|\mathbf{u})$. If $\mathrm{Tr}(\mathbf{K_{XX}} - \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}}) = 0$, the inducing variables become sufficient statistics and can reproduce exactly the GP prediction.

Hensman and Lawrence (2014) further corroborated this argument from an infor-

mation theoretic perspective. In particular, the conditional entropy[2] of the training conditional is given by:

$$H(\mathbf{f}|\mathbf{u}) = \frac{1}{2}\log|\mathbf{K_{XX}} - \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}}|. \tag{2.39}$$

which intuitively represents the amount of additional information we need to specify about the distribution of $\mathbf{f}$ given that we already have inducing variables $\mathbf{u}$. Hensman and Lawrence (2014) argued that a small trace term leads to a small log determinant and consequently results in smaller conditional entropy. As a result, a smaller conditional entropy implies smaller required amount of additional information to specify the distribution of $\mathbf{f}$ and therefore increased informativeness of the inducing variables.

2.4 Stochastic Variational Sparse Gaussian Process

Note that the computational complexity for the EBLO in (2.37) is still identical to that of DTC, FITC, PITC, and PIC. As a result, variational SGP incurs linear time in the data size which is still prohibitively expensive for training with million-sized datasets. To this end, Hensman *et al.* (2013) proposed a novel stochastic variational inference (SVI) method [Hoffman *et al.*, 2013] for GP whose ELBO factorizes with respect to data points, and hence incurs constant time complexity which conveniently scales to million-sized datasets. Instead of marginalizing out the inducing variables $\mathbf{u}$ in the ELBO in (2.37), the work of [Hensman *et al.*, 2013] maintains an explicit representation of the inducing variables, and derives a lower bound which includes an explicit variational distribution for $q(\mathbf{u})$.

As we have mentioned in Section 2.3, the optimal $q^*(\mathbf{u})$ is Gaussian, therefore, the work of [Hensman *et al.*, 2013] treats $q(\mathbf{u})$ as variational parameters: $q(\mathbf{u}) = \mathcal{N}(\mathbf{m}, \mathbf{S})$.

[2]Conditional entropy quantifies the amount of information needed to describe the outcome of a random variable Y given that the value of another random variable X is known, denoted as $H(Y|X)$.

Then the new EBLO becomes:

$$\begin{aligned}\mathcal{L}_{SVI}(q) = \log \mathcal{N}(\mathbf{y}|\mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{m}, \sigma_n^2\mathbf{I}) - \frac{1}{2\sigma_n^2}\mathrm{Tr}(\mathbf{K_{XX}} - \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}}) \\ -\frac{1}{2\sigma_n^2}\mathrm{Tr}(\mathbf{S}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}}\mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}) - \mathrm{KL}(q(\mathbf{u})||p(\mathbf{u}))\end{aligned} \tag{2.40}$$

Interestingly, the ELBO above can factorize over data points:

$$\begin{aligned}\mathcal{L}_{SVI}(q) = \sum_{n=1}^{N}\{\log \mathcal{N}(y_n|\mathbf{K}_{\mathbf{x}_n\mathbf{Z}}\mathbf{K_{ZZ}^{-1}}\mathbf{m}, \sigma_n^2) - \frac{1}{2\sigma_n^2}\mathrm{Tr}(\mathbf{K}_{\mathbf{x}_n\mathbf{x}_n} - \mathbf{K}_{\mathbf{x}_n\mathbf{Z}}\mathbf{K_{ZZ}^{-1}}\mathbf{K}_{\mathbf{Z}\mathbf{x}_n}) \\ -\frac{1}{2\sigma_n^2}\mathrm{Tr}(\mathbf{S}\mathbf{K_{ZZ}^{-1}}\mathbf{K}_{\mathbf{Z}\mathbf{x}_n}\mathbf{K}_{\mathbf{x}_n\mathbf{Z}}\mathbf{K_{ZZ}^{-1}})\} - \mathrm{KL}(q(\mathbf{u})||p(\mathbf{u}))\end{aligned} \tag{2.41}$$

The key property of $\mathcal{L}_{SVI}(q)$ is that the ELBO can be written as a sum of N terms, and each term corresponds to one input-output pair $\{\mathbf{x}_n, y_n\}$. Thus, it allows the use of the stochastic optimization (stochastic gradient method) when training $q(\mathbf{u})$. Another interesting property of this algorithm can be observed by inspecting the gradients. The gradients of $\mathcal{L}_{SVI}(q)$ with respect to the variational parameters of $q(\mathbf{u})$ are:

$$\begin{aligned}\frac{\partial \mathcal{L}_{SVI}}{\partial \mathbf{m}} &= \sigma_n^{-2}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}}\mathbf{y} - \sigma_n^{-2}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}}\mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{m} \\ \frac{\partial \mathcal{L}_{SVI}}{\partial \mathbf{S}} &= \frac{1}{2}\mathbf{S}^{-1} - \frac{1}{2\sigma_n^2}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}}\mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}.\end{aligned} \tag{2.42}$$

Setting the derivatives to zero recovers the optimal $q(\mathbf{u})$ which turns out to be identical to that of (2.36). Therefore, it is guaranteed to achieve asymptotic convergence to the optimal variational parameters.

2.5 Summary

The focus of this thesis is to develop novel approximate inference methods for inducing variables-based GP and DGP models specifically[3].

As shown in Section 2.4, the stochastic variational SGP is the first work to scale GP to million-sized datasets. However, its predictive performance is severely compromised by the restrictive DTC approximation. Subsequently, a unifying framework of variational SGP models are proposed by Hoang *et al.* (2015) to perform SVI for any SGP model spanned by the unifying view elaborated in Section 2.2. Nevertheless, the variational SGP models and their stochastic variants suffer from certain critical issues which will be explained concretely in Chapter 3, which motivates us to propose a novel variational Bayesian SGP model.

[3]An alternative to inducing variables-based methods is to directly modify the covariance matrix, e.g. [Gal and Turner, 2015], [Hoang *et al.*, 2017] and [Cutajar *et al.*, 2017].

Chapter 3

Bayesian Sparse Gaussian Process Regression

This chapter is based on the paper published in the 2019 International Joint Conference on Neural Networks (IJCNN-19): "Stochastic Variational Inference for Bayesian Sparse Gaussian Process Regression" [Yu *et al.*, 2019b].

3.1 Introduction

In the previous chapter, we briefly mentioned that the variational SGP models and their stochastic variants face certain issues. Specifically, they suffer from the following critical issues: (a) The equivalence of maximizing the ELBO and minimizing the KL distance only holds for the case of fixed hyperparameters; otherwise, since the log-marginal likelihood also depends on the same hyperparameters that are optimized to maximize its ELBO, the resulting KL distance, which quantifies the gap between the log-marginal likelihood and its ELBO, is not guaranteed to be minimized; (b) similar to variational expectation-maximization [Wainwright and Jordan, 2008], the log-marginal likelihood does not necessarily increase in each iteration of gradient

ascent to refine the hyperparameter estimates to improve its ELBO; and (c) they all find point estimates of the hyperparameters, which risks overfitting, especially when the number of hyperparameters is large.

To resolve these issues, the notable work of [Titsias and Lázaro-Gredilla, 2013] has introduced a *variational Bayesian DTC* (VBDTC) approximation (Section 3.5) capable of learning a variational distribution of the hyperparameters. This learned distribution of hyperparameters is particularly desirable in conveying the uncertainty/confidence of the hyperparameter estimates and for use in Bayesian GP regression (Section 3.7), active learning [Cao *et al.*, 2013; Hoang *et al.*, 2014; Low *et al.*, 2008; Low *et al.*, 2009; Low *et al.*, 2011; Ouyang *et al.*, 2014; Zhang *et al.*, 2016], Bayesian optimization [Daxberger and Low, 2017; Hernández-Lobato *et al.*, 2014; Hoang *et al.*, 2018; Ling *et al.*, 2016], among others. Unfortunately, such a VBDTC approximation cannot handle big data (e.g., million-sized datasets) because it incurs linear time in the data size per iteration of gradient ascent.

It remains an open question whether more refined SGP regression models such as FITC, PITC, and PIC as we have discussed in Chapter 2 are amenable to variational Bayesian treatment and able to achieve scalability through stochastic optimization.

To address this question, this chapter presents a novel VI framework for deriving a family of Bayesian SGP regression models (e.g., VBDTC, VBFITC, VBPIC) whose approximations are, interestingly, variationally optimal with respect to the GP regression model enriched with various corresponding correlation structures of the observation noises (Section 3.5). Our framework introduces a novel reparameterization of the GP model (Section 3.4) for enabling a variational treatment of the distribution of hyperparameters. Unlike VBDTC, our framework does not need to assume independently distributed observation noises with constant variance and is thus more robust to different noise correlation structures, hence catering to more realistic applications of GP. Furthermore, instead of just considering the distribu-

tion of hyperparameters as variational parameters [Titsias and Lázaro-Gredilla, 2013; Gal and Turner, 2015], we jointly treat both the distributions of the inducing variables and hyperparameters as variational parameters, which enables the decomposability of the ELBO that in turn can be exploited for stochastic optimization (Section 3.6). Such a stochastic optimization involves iteratively following the stochastic gradient of the ELBO to improve its estimates of the optimal variational distributions of the inducing variables and hyperparameters (and hence the predictive distribution (Section 3.7)) of our *variational Bayesian SGP* (VBSGP) regression models and is guaranteed to achieve asymptotic convergence to them. We show that the derived stochastic gradient is an unbiased estimator of the exact gradient and can be computed in constant time (i.e., independent of data size) per iteration, thus achieving scalability to big data. We empirically evaluate the performance of the stochastic variants of our VBSGP regression models on two real-world datasets (Section 3.8).

3.2 Notations

With a slight abuse of notations, In this chapter we let $\mathcal{X}$ denote a d-dimensional input feature space such that each input vector $\mathbf{x} \in \mathcal{X}$ is associated with a latent output variable $f_{\mathbf{x}}$. Let $\{f_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{X}}$ denote a GP. Then, the SE kernel function is denoted as: $k_{\mathbf{x}\mathbf{x}'} \triangleq \sigma_f^2 \exp(-0.5\|\mathbf{\Lambda}\mathbf{x} - \mathbf{\Lambda}\mathbf{x}'\|_2^2)$ where $\mathbf{\Lambda} = \mathrm{diag}[\lambda_1, \ldots, \lambda_d]$ and σ_f^2 are its *inverted* length-scale and signal variance hyperparameters, respectively. $\boldsymbol{\theta} \triangleq \{\lambda_1, \ldots, \lambda_d, \sigma_f\}^\top$ denotes the the hyperparameters.

Suppose that a column vector $\mathbf{y}_\mathcal{D} \triangleq (y_{\mathbf{x}})_{\mathbf{x}\in\mathcal{D}}^\top$ of noisy observed outputs $y_{\mathbf{x}} \triangleq f_{\mathbf{x}} + \varepsilon_{\mathbf{x}}$ (i.e., corrupted by an additive noise $\varepsilon_{\mathbf{x}}$) is available for some set $\mathcal{D} \subset \mathcal{X}$ of training inputs such that $\boldsymbol{\varepsilon}_\mathcal{D} \triangleq (\varepsilon_{\mathbf{x}})_{\mathbf{x}\in\mathcal{D}}^\top$ follows a multivariate Gaussian distribution $p(\boldsymbol{\varepsilon}_\mathcal{D}) \triangleq \mathcal{N}(\mathbf{0}, \mathbf{C}_{\mathcal{D}\mathcal{D}})$ where $\mathbf{C}_{\mathcal{D}\mathcal{D}}$ is a covariance matrix representing the correlation of observation noises $\boldsymbol{\varepsilon}_\mathcal{D}$. Note that unlike the observation noises in Section 1.1.4

which is a diagonal matrix $\mathbf{C}_{\mathcal{D}\mathcal{D}} = \sigma_n^2\mathbf{I}$. $\mathbf{C}_{\mathcal{D}\mathcal{D}}$ can be more general which incorporates the correlation among different observations. Let $\mathbf{f}_{\mathcal{U}} \triangleq (f_{\mathbf{x}})_{\mathbf{x}\in\mathcal{U}}^{\top}$ denote the inducing output variables for some small set $\mathcal{U} \subset \mathcal{X}$ of inducing inputs where $|\mathcal{U}| \ll |\mathcal{D}|$. PIC stands for the most refined SGP regression model among all the SGP regression models we have discussed in Section 2.2, without loss of generality, we adopt the same assumptions such that $\mathbf{f}_{\mathcal{D}}$ factorizes over a pre-defined partition of the input space $\mathcal{X}$ into B disjoint subsets $\mathcal{X}_1, \ldots, \mathcal{X}_B$ (i.e., $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \ldots \cup \mathcal{X}_B$). Formally, suppose $\mathbf{x}^\star \in \mathcal{X}_B$, then the joint distribution for training and test conditional is:

$$p(f_{\mathbf{x}^*}, \mathbf{f}_{\mathcal{D}}|\mathbf{f}_{\mathcal{U}}) = p(f_{\mathbf{x}^*}|\mathbf{f}_{\mathcal{D}_B}, \mathbf{f}_{\mathcal{U}}) \prod_{i=1}^{B} p(\mathbf{f}_{\mathcal{D}_i}|\mathbf{f}_{\mathcal{U}}) \tag{3.1}$$

where $\mathbf{f}_{\mathcal{D}_i} \triangleq (f_{\mathbf{x}})_{\mathbf{x}\in\mathcal{D}_i}^{\top}$ is a column vector of latent outputs for the disjoint subset $\mathcal{D}_i \triangleq (\mathcal{X}_i \cap \mathcal{D}) \subset \mathcal{D}$ for $i = 1, 2, \ldots, B$. Using this factorization, the predictive distribution can be represented as:

$$p(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}}) = \int p(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}_B}, \mathbf{f}_{\mathcal{U}})\ p(\mathbf{f}_{\mathcal{U}}|\mathbf{y}_{\mathcal{D}})\ \mathrm{d}\mathbf{f}_{\mathcal{U}} \simeq \int q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}_B}, \mathbf{f}_{\mathcal{U}})\ q(\mathbf{f}_{\mathcal{U}})\ \mathrm{d}\mathbf{f}_{\mathcal{U}} \tag{3.2}$$

where $\mathbf{y}_{\mathcal{D}_B} \triangleq (y_{\mathbf{x}})_{\mathbf{x}\in\mathcal{D}_B}^{\top}$ is a vector of noisy observed outputs for the subset $\mathcal{D}_B$ of training inputs.

3.3 Bayesian SGP Regression Models

For the variational SGP regression models described in Section 2.3, point estimates of their hyperparameters are learned, which is vulnerable to overfitting, especially when the number of hyperparameters is large. To mitigate this issue of overfitting, a Bayesian approach to SGP regression can be employed by introducing priors $p(\boldsymbol{\theta}) \triangleq p(\boldsymbol{\Lambda})\ p(\sigma_f)$ over hyperparameters $\boldsymbol{\theta}$, thus yielding the predictive distribution:

$$
\begin{aligned}
p(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}}) &= \int p(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}_B}, \mathbf{f}_{\mathcal{U}}, \boldsymbol{\theta})\ p(\mathbf{f}_{\mathcal{U}}, \boldsymbol{\theta}|\mathbf{y}_{\mathcal{D}})\ \mathrm{d}\mathbf{f}_{\mathcal{U}}\ \mathrm{d}\boldsymbol{\theta} \\
&\simeq \int q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}_B}, \mathbf{f}_{\mathcal{U}}, \boldsymbol{\theta})\ q(\mathbf{f}_{\mathcal{U}}, \boldsymbol{\theta})\ \mathrm{d}\mathbf{f}_{\mathcal{U}}\ \mathrm{d}\boldsymbol{\theta}
\end{aligned}
\tag{3.3}
$$

where $p(\mathbf{f}_{\mathcal{U}}, \boldsymbol{\theta}|\mathbf{y}_{\mathcal{D}})$ is approximated by $q(\mathbf{f}_{\mathcal{U}}, \boldsymbol{\theta})$ which generalizes $q(\mathbf{f}_{\mathcal{U}})$ above by additionally and jointly considering the hyperparameters $\boldsymbol{\theta}$ as variational variables. Though (3.3), in principle, allows a Bayesian treatment of $\boldsymbol{\theta}$ to be incorporated into the existing SGP regression models, computing the resulting predictive distribution is intractable because it involves integrating, over $\boldsymbol{\Lambda}$, probability terms in (3.3) containing the inverse of $\mathbf{K}_{\mathcal{U}\mathcal{U}} \triangleq (k_{\mathbf{x}\mathbf{x}'})_{\mathbf{x},\mathbf{x}'\in\mathcal{U}}$ that depends on $\boldsymbol{\Lambda}$ but without an analytical form with respect to $\boldsymbol{\Lambda}$. To resolve this, we introduce a reparameterization trick to make the prior distribution of inducing outputs independent of the hyperparameters $\boldsymbol{\theta}$, as discussed next.

3.4 Reparameterizing Bayesian SGP Regression Models

Let $\phi : \mathbb{R}^d \to \mathcal{H}$ denote a non-linear feature map from the input space $\mathbb{R}^d$ into a *reproducing kernel Hilbert space* (RKHS) $\mathcal{H}$ whose inner product is defined as

$$
\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}} \triangleq \exp(-0.5\|\mathbf{x} - \mathbf{x}'\|_2^2). \tag{3.4}
$$

Given ϕ, the GP covariance/kernel function can be interpreted as

$$
k_{\mathbf{x}\mathbf{x}'} \triangleq \langle \sigma(\mathbf{x})\phi(\boldsymbol{\Lambda}\mathbf{x}), \sigma(\mathbf{x}')\phi(\boldsymbol{\Lambda}\mathbf{x}') \rangle_{\mathcal{H}} = \sigma(\mathbf{x})\sigma(\mathbf{x}') \exp(-0.5\|\boldsymbol{\Lambda}\mathbf{x} - \boldsymbol{\Lambda}\mathbf{x}'\|_2^2) \tag{3.5}
$$

where σ is an arbitrary function mapping from $\mathbb{R}^d$ to $\mathbb{R}$. This implies $k_{\mathbf{x}\mathbf{x}} = \sigma^2(\mathbf{x})$ which allows $\sigma^2(\mathbf{x})$ to be interpreted as the prior variance $k_{\mathbf{x}\mathbf{x}}$ of $f_{\mathbf{x}}$ (Section 3.2).

We will now describe the reparameterization trick: Let $\mathcal{I} \triangleq \{\mathbf{\Lambda x}\}_{\mathbf{x}\in\mathcal{U}}$. Intuitively, $\mathcal{I}$ can be interpreted as a set of *rotated inducing inputs* with the diagonal matrix $\mathbf{\Lambda}$ of inverted length-scales being the rotation matrix. Let each rotated inducing input $\mathbf{z} \in \mathcal{I}$ be associated with a latent output variable $s_{\mathbf{z}}$. By definition of RKHS, the covariance is:

$$\text{cov}[s_{\mathbf{z}}, s_{\mathbf{z}'}] \triangleq \langle \sigma(\mathbf{z})\phi(\mathbf{z}), \sigma(\mathbf{z}')\phi(\mathbf{z}')\rangle_{\mathcal{H}} = \sigma(\mathbf{z})\sigma(\mathbf{z}') \exp(-0.5\|\mathbf{z} - \mathbf{z}'\|_2^2) \tag{3.6}$$

for all $\mathbf{z}, \mathbf{z}' \in \mathcal{I}$. By assuming that the prior variances of $s_{\mathbf{z}}$ for all $\mathbf{z} \in \mathcal{I}$ are identical and equal to some constant ζ^2 (i.e., $\sigma(\mathbf{z}) = \zeta > 0$):

$$\text{cov}[s_{\mathbf{z}}, s_{\mathbf{z}'}] = \zeta^2 \exp(-0.5\|\mathbf{z} - \mathbf{z}'\|_2^2) \tag{3.7}$$

which is independent of $\boldsymbol{\theta}$. Consequently, the prior covariance matrix

$$\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}} \triangleq (\text{cov}[s_{\mathbf{z}}, s_{\mathbf{z}'}])_{\mathbf{z},\mathbf{z}'\in\mathcal{I}}$$

of the inducing output variables $\mathbf{s}_{\mathcal{I}} \triangleq (s_{\mathbf{z}})_{\mathbf{z}\in\mathcal{I}}^{\top}$ is independent of $\boldsymbol{\theta}$. Furthermore, the cross-covariance matrix $\mathbf{K}_{\mathcal{D}\mathcal{I}} \triangleq (\text{cov}[f_{\mathbf{x}}, s_{\mathbf{z}}])_{\mathbf{x}\in\mathcal{D},\mathbf{z}\in\mathcal{I}}$ between the latent outputs $\mathbf{f}_{\mathcal{D}}$ for some set $\mathcal{D}$ of training inputs and the inducing outputs $\mathbf{s}_{\mathcal{I}}$ can be computed analytically using the definition of RKHS:

$$\text{cov}[f_{\mathbf{x}}, s_{\mathbf{z}}] = \langle \sigma(\mathbf{x})\phi(\mathbf{\Lambda x}), \sigma(\mathbf{z})\phi(\mathbf{z})\rangle_{\mathcal{H}} = \zeta\sigma(\mathbf{x}) \exp(-0.5\|\mathbf{\Lambda x} - \mathbf{z}\|_2^2). \tag{3.8}$$

Like many existing GP models, the prior variances of $f_{\mathbf{x}}$ for all $\mathbf{x} \in \mathcal{X}$ are assumed to be identical and equal to a signal variance hyperparameter σ_f^2 (i.e., $\sigma(\mathbf{x}) = \sigma_f$) for tractable learning, hence circumventing the need to learn an infinite number of prior variance hyperparameters. The resulting representation of the GP model from the

reparameterization trick will allow the optimal variational distributions of inducing outputs $\mathbf{s}_{\mathcal{I}}$ and hyperparameters $\boldsymbol{\theta}$ (hence the predictive distribution) to be tractably derived for a family of VBSGP regression models, as discussed in Section 3.5.

Remark 1. The definition of $\mathcal{I}$ seems to suggest its construction by first selecting the inducing inputs $\mathcal{U}$ and then rotating them via $\boldsymbol{\Lambda}$, which is not possible since $\boldsymbol{\Lambda}$ is not known *a priori*. However, as shall be discussed in Section 3.5, it is possible to first select $\mathcal{I}$ and then optimize the variational distribution of $\boldsymbol{\Lambda}$, which has an effect of optimizing the distribution of inducing inputs $\mathcal{U}$ in original input space $\mathcal{X}$.

Remark 2. Let $\mathcal{Z} \triangleq \{\boldsymbol{\Lambda}\mathbf{x}\}_{\mathbf{x}\in\mathcal{X}}$. By setting the (identical) prior variances of $s_{\mathbf{z}}$ for all $\mathbf{z} \in \mathcal{Z}$ to unity (i.e., $\zeta = 1$), $\{s_{\mathbf{z}}\}_{\mathbf{z}\in\mathcal{Z}}$ denote a *standard* GP with unit signal variance and length-scales [Titsias and Lázaro-Gredilla, 2013], which is a special case of our representation of the GP model here. Then, $f_{\mathbf{x}} = \sigma_f s_{\boldsymbol{\Lambda}\mathbf{x}}$ for all $\mathbf{x} \in \mathcal{X}$.

3.5 Variational Bayesian SGP Regression Models

Using our representation of the GP model defined above (Section 3.4), the predictive distribution (3.3) of a Bayesian SGP regression model can be slightly modified to

$$p(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}}) = \int p(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}_B}, \mathbf{s}_{\mathcal{I}}, \boldsymbol{\theta})\ p(\mathbf{s}_{\mathcal{I}}, \boldsymbol{\theta}|\mathbf{y}_{\mathcal{D}})\ \mathrm{d}\mathbf{s}_{\mathcal{I}}\ \mathrm{d}\boldsymbol{\theta} \tag{3.9}$$

such that deriving the posterior $p(\mathbf{s}_{\mathcal{I}}, \boldsymbol{\theta}|\mathbf{y}_{\mathcal{D}}) = p(\mathbf{y}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \boldsymbol{\theta})/p(\mathbf{y}_{\mathcal{D}})$ requires computing the likelihood:

$$p(\mathbf{y}_{\mathcal{D}}) = \mathbb{E}_{\boldsymbol{\theta}}\left[\int p(\mathbf{y}_{\mathcal{D}}|\mathbf{f}_{\mathcal{D}})p(\mathbf{f}_{\mathcal{D}}|\mathbf{s}_{\mathcal{I}}, \boldsymbol{\theta})p(\mathbf{s}_{\mathcal{I}})\mathrm{d}\mathbf{f}_{\mathcal{D}}\mathrm{d}\mathbf{s}_{\mathcal{I}}\right] \tag{3.10}$$

where $p(\boldsymbol{\theta}) \triangleq \mathcal{N}(\mathbf{1}, \mathrm{diag}[0.1])$, $p(\mathbf{s}_{\mathcal{I}}) = \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}})$, $p(\mathbf{y}_{\mathcal{D}}|\mathbf{f}_{\mathcal{D}}) = \mathcal{N}(\mathbf{f}_{\mathcal{D}}, \mathbf{C}_{\mathcal{D}\mathcal{D}})$, and

$$p(\mathbf{f}_{\mathcal{D}}|\mathbf{s}_{\mathcal{I}},\boldsymbol{\theta}) = \mathcal{N}(\mathbf{K}_{\mathcal{D}\mathcal{I}}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{s}_{\mathcal{I}}, \mathbf{K}_{\mathcal{D}\mathcal{D}} - \mathbf{K}_{\mathcal{D}\mathcal{I}}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{K}_{\mathcal{I}\mathcal{D}}) \tag{3.11}$$

such that $\mathbf{K}_{\mathcal{D}\mathcal{I}}$ is previously defined in Section 3.4 and $\mathbf{K}_{\mathcal{I}\mathcal{D}} = \mathbf{K}_{\mathcal{D}\mathcal{I}}^{\top}$. However, the integration in (3.10) (and hence $p(\mathbf{s}_{\mathcal{I}},\boldsymbol{\theta}|\mathbf{y}_{\mathcal{D}})$) cannot be evaluated in closed form. To resolve this, instead of using exact inference, we adopt variational inference to approximate the posterior distribution

$$p(\mathbf{f}_{\mathcal{D}},\mathbf{s}_{\mathcal{I}},\boldsymbol{\theta}|\mathbf{y}_{\mathcal{D}}) = p(\mathbf{f}_{\mathcal{D}}|\mathbf{s}_{\mathcal{I}},\boldsymbol{\theta})\ p(\mathbf{s}_{\mathcal{I}},\boldsymbol{\theta}|\mathbf{y}_{\mathcal{D}})$$

with a factorized variational distribution

$$q(\mathbf{f}_{\mathcal{D}},\mathbf{s}_{\mathcal{I}},\boldsymbol{\theta}) \triangleq p(\mathbf{f}_{\mathcal{D}}|\mathbf{s}_{\mathcal{I}},\boldsymbol{\theta})\ q(\mathbf{s}_{\mathcal{I}})\ q(\boldsymbol{\theta})$$

where $p(\mathbf{f}_{\mathcal{D}}|\mathbf{s}_{\mathcal{I}},\boldsymbol{\theta})$ is the exact training conditional (3.11),

$$q(\mathbf{s}_{\mathcal{I}}) \triangleq \mathcal{N}(\mathbf{m},\mathbf{S}),\ q(\boldsymbol{\theta}) \triangleq q(\boldsymbol{\Lambda})\ q(\sigma_f),\ q(\sigma_f) \triangleq \mathcal{N}(\alpha,\beta)$$
$$q(\boldsymbol{\Lambda}) \triangleq \prod_{i=1}^{d} \mathcal{N}(\lambda_i|\nu_i,\xi_i) \text{ with } \boldsymbol{\nu} \triangleq (\nu_1,\ldots,\nu_d)^{\top} \text{ and } \boldsymbol{\Xi} \triangleq \operatorname{diag}[\xi_1,\ldots,\xi_d].$$

Then, the log-marginal likelihood $\log p(\mathbf{y}_{\mathcal{D}})$ can be decomposed into a sum of its ELBO $\mathcal{L}(q)$ and the nonnegative KL distance between the variational distribution $q(\mathbf{f}_{\mathcal{D}},\mathbf{s}_{\mathcal{I}},\boldsymbol{\theta})$ and the posterior distribution $p(\mathbf{f}_{\mathcal{D}},\mathbf{s}_{\mathcal{I}},\boldsymbol{\theta}|\mathbf{y}_{\mathcal{D}})$, the latter of which quantifies the gap between $\log p(\mathbf{y}_{\mathcal{D}})$ and $\mathcal{L}(q)$, that is,

$$\log p(\mathbf{y}_{\mathcal{D}}) = \mathcal{L}(q) + \mathrm{KL}(q(\mathbf{f}_{\mathcal{D}},\mathbf{s}_{\mathcal{I}},\boldsymbol{\theta})||p(\mathbf{f}_{\mathcal{D}},\mathbf{s}_{\mathcal{I}},\boldsymbol{\theta}|\mathbf{y}_{\mathcal{D}})), \tag{3.12}$$

as derived in Appendix B.1 where

$$\mathcal{L}(q) \triangleq \int q(\mathbf{f}_{\mathcal{D}},\mathbf{s}_{\mathcal{I}},\boldsymbol{\theta}) \log \frac{p(\mathbf{y}_{\mathcal{D}},\mathbf{f}_{\mathcal{D}},\mathbf{s}_{\mathcal{I}},\boldsymbol{\theta})}{q(\mathbf{f}_{\mathcal{D}},\mathbf{s}_{\mathcal{I}},\boldsymbol{\theta})}\ \mathrm{d}\mathbf{f}_{\mathcal{D}}\ \mathrm{d}\mathbf{s}_{\mathcal{I}}\ \mathrm{d}\boldsymbol{\theta}. \tag{3.13}$$

Remark 3. The likelihood term $p(\mathbf{y}_\mathcal{D})$ (3.10) in (3.12) is a constant with respect to $q(\mathbf{s}_\mathcal{I})$ and $q(\boldsymbol{\theta})$ (specifically, their parameters $\mathbf{m}, \mathbf{S}, \boldsymbol{\nu}, \boldsymbol{\Xi}, \alpha, \beta$). Consequently,

$$\max_{q(\mathbf{f}_\mathcal{D}, \mathbf{s}_\mathcal{I}, \boldsymbol{\theta})} \mathcal{L}(q(\mathbf{f}_\mathcal{D}, \mathbf{s}_\mathcal{I}, \boldsymbol{\theta})) = \min_{q(\mathbf{f}_\mathcal{D}, \mathbf{s}_\mathcal{I}, \boldsymbol{\theta})} \mathrm{KL}(q(\mathbf{f}_\mathcal{D}, \mathbf{s}_\mathcal{I}, \boldsymbol{\theta}) || p(\mathbf{f}_\mathcal{D}, \mathbf{s}_\mathcal{I}, \boldsymbol{\theta} | \mathbf{y}_\mathcal{D})).$$

This equivalence, however, does not hold for existing variational SGP regression models and their stochastic and distributed variants optimizing point estimates of all hyperparameters, as discussed in Section 3.1.

The variational inference framework of [Titsias and Lázaro-Gredilla, 2013] is similar in spirit to the above. However, the framework of [Titsias and Lázaro-Gredilla, 2013] assumes i.i.d. observation noises (i.e., $\mathbf{C}_{\mathcal{D}\mathcal{D}} = \sigma_n^2 \mathbf{I}$ and $\zeta = 1$) and ignores their correlation, which consequently yields the VBDTC approximation (see Remark 4 later). The challenge remains in investigating whether the other more refined SGP regression models spanned by the unifying view of [Quiñonero-Candela and Rasmussen, 2005] (e.g., FITC, PITC, PIC) are amenable to such a variational Bayesian treatment since they have been empirically demonstrated [Hoang *et al.*, 2015; Hoang *et al.*, 2016] to give better predictive performance than DTC.

To address this challenge, our key idea is to relax the strong assumption of i.i.d. observation noises with constant variance σ_n^2 imposed by VBDTC and allow observation noises to be correlated with some structure across the input space, hence being robust to different noise correlation structures and in turn catering to more realistic applications of GP. Interestingly, this results in a noise-robust family of *variational Bayesian SGP* (VBSGP) regression models (e.g., VBDTC, VBFITC, VBPIC), which we will describe below.

Let $\mathbf{C}_{\mathcal{D}\mathcal{D}} \triangleq \mathrm{blkdiag}[\mathbf{K}^\varepsilon_{\mathcal{D}\mathcal{D}} - \mathbf{K}^\varepsilon_{\mathcal{D}\mathcal{U}} \mathbf{K}^{\varepsilon^{-1}}_{\mathcal{U}\mathcal{U}} \mathbf{K}^\varepsilon_{\mathcal{U}\mathcal{D}}] + \sigma_n^2 \mathbf{I}$ be a block-diagonal noise covariance matrix constructed from the B diagonal blocks of $\mathbf{K}^\varepsilon_{\mathcal{D}\mathcal{D}} - \mathbf{K}^\varepsilon_{\mathcal{D}\mathcal{U}} \mathbf{K}^{\varepsilon^{-1}}_{\mathcal{U}\mathcal{U}} \mathbf{K}^\varepsilon_{\mathcal{U}\mathcal{D}} + \sigma_n^2 \mathbf{I}$, each of which is a matrix $\mathbf{C}_{\mathcal{D}_i\mathcal{D}_i} \triangleq \mathbf{K}^\varepsilon_{\mathcal{D}_i\mathcal{D}_i} - \mathbf{K}^\varepsilon_{\mathcal{D}_i\mathcal{U}} \mathbf{K}^{\varepsilon^{-1}}_{\mathcal{U}\mathcal{U}} \mathbf{K}^\varepsilon_{\mathcal{U}\mathcal{D}_i} + \sigma_n^2 \mathbf{I}$ for $i = 1, \ldots, B$,

and $\mathbf{K}^{\varepsilon}_{\mathcal{D}\mathcal{D}} \triangleq (k^{\varepsilon}_{\mathbf{x}\mathbf{x}'})_{\mathbf{x},\mathbf{x}'\in\mathcal{D}}$, $\mathbf{K}^{\varepsilon}_{\mathcal{D}\mathcal{U}} \triangleq (k^{\varepsilon}_{\mathbf{x}\mathbf{x}'})_{\mathbf{x}\in\mathcal{D},\mathbf{x}'\in\mathcal{U}}$, $\mathbf{K}^{\varepsilon}_{\mathcal{U}\mathcal{U}} \triangleq (k^{\varepsilon}_{\mathbf{x}\mathbf{x}'})_{\mathbf{x},\mathbf{x}'\in\mathcal{U}}$, and $\mathbf{K}^{\varepsilon}_{\mathcal{U}\mathcal{D}} \triangleq \mathbf{K}^{\varepsilon^\top}_{\mathcal{D}\mathcal{U}}$ are matrices with components $k^{\varepsilon}_{\mathbf{x}\mathbf{x}'}$ defined by a covariance function similar to that used for $k_{\mathbf{x}\mathbf{x}'}$ (Section 3.2) but with different hyperparameter values[1]. Our first major result ensues:

Theorem 1. *$\mathcal{L}(q)$ in (3.12) can be analytically evaluated as*

$$\mathcal{L}(q) = \frac{1}{2}\Big(2\mathbf{m}^\top\boldsymbol{\Sigma}^{-1}_{\mathcal{I}\mathcal{I}}\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}}\mathbf{C}^{-1}_{\mathcal{D}\mathcal{D}}\mathbf{y}_{\mathcal{D}} - \mathbf{m}^\top\mathbf{Q}\mathbf{m} - \mathrm{Tr}[\mathbf{S}\mathbf{Q}] - \mathrm{Tr}[\mathbf{C}^{-1}_{\mathcal{D}\mathcal{D}}\boldsymbol{\Upsilon}_{\mathcal{D}\mathcal{D}}] + \mathrm{Tr}[\boldsymbol{\Sigma}^{-1}_{\mathcal{I}\mathcal{I}}\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}] + \log|\mathbf{S}| - \boldsymbol{\nu}^\top\boldsymbol{\nu} - \mathrm{Tr}[\boldsymbol{\Xi}] + \log|\boldsymbol{\Xi}| - \alpha^2 - \beta + \log\beta\Big) + \mathrm{const} \tag{3.14}$$

where $\mathbf{Q} \triangleq \boldsymbol{\Sigma}^{-1}_{\mathcal{I}\mathcal{I}}\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}\boldsymbol{\Sigma}^{-1}_{\mathcal{I}\mathcal{I}} + \boldsymbol{\Sigma}^{-1}_{\mathcal{I}\mathcal{I}}$. More interestingly, using the above expression, it can be shown that $\mathcal{L}(q)$ is maximized at $q^*(\mathbf{s}_{\mathcal{I}}) = \mathcal{N}(\mathbf{m}^*, \mathbf{S}^*)$ where

$$\begin{aligned}\mathbf{m}^* &\triangleq \boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}(\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}} + \boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}})^{-1}\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}}\mathbf{C}^{-1}_{\mathcal{D}\mathcal{D}}\mathbf{y}_{\mathcal{D}}\ ,\\ \mathbf{S}^* &\triangleq \boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}(\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}} + \boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}})^{-1}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}\end{aligned} \tag{3.15}$$

such that $\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}} \triangleq \mathbb{E}_{q(\boldsymbol{\theta})}[\mathbf{K}_{\mathcal{I}\mathcal{D}}]$, $\boldsymbol{\Upsilon}_{\mathcal{D}\mathcal{D}} \triangleq \mathbb{E}_{q(\boldsymbol{\theta})}[\mathbf{K}_{\mathcal{D}\mathcal{D}}]$, $\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}} \triangleq \mathbb{E}_{q(\boldsymbol{\theta})}[\mathbf{K}_{\mathcal{I}\mathcal{D}}\mathbf{C}^{-1}_{\mathcal{D}\mathcal{D}}\mathbf{K}_{\mathcal{D}\mathcal{I}}]$, and const *absorbs all terms indep. of $\mathbf{m}$, $\mathbf{S}$, $\boldsymbol{\nu}$, $\boldsymbol{\Xi}, \alpha, \beta$.*

Its proof is in Appendix B.2. Appendix B.4.1 gives the closed-form expressions of $\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}}$, $\boldsymbol{\Upsilon}_{\mathcal{D}\mathcal{D}}$, and $\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}$.

Remark 4. Note that $q^*(\mathbf{s}_{\mathcal{I}})$ in Theorem 1 closely resembles that of PIC and PITC (see eqs. 64 and 65 in Appendix D.1.1 of [Hoang *et al.*, 2015]) except for the expectation over hyperparameters $\boldsymbol{\theta}$ due to the variational Bayesian treatment. So, we call them

[1]We do not assign any prior over the hyperparameters of $k^{\varepsilon}_{\mathbf{x}\mathbf{x}'}$ and the noise variance σ_n^2. Instead, they are treated as parameters optimized to maximize $\mathcal{L}(q)$ via stochastic gradient ascent [Hensman *et al.*, 2013]. In our experiments, we observe that even if we set the hyperparameters of $k^{\varepsilon}_{\mathbf{x}\mathbf{x}'}$ by hand, the predictive performance does not vary much and our VBPIC approximation can significantly outperform the state-of-the-art variational SGP regression models and their stochastic and distributed variants. A Bayesian treatment of these hyperparameters is highly non-trivial due to a complication similar to that discussed in Section 3.3 and will be investigated in our future work.

VBPIC and VBPITC, respectively. By setting $B = |\mathcal{D}|$, $\mathbf{C}_{\mathcal{D}\mathcal{D}}$ becomes a diagonal matrix to give VBFIC and VBFITC. When $\mathbf{C}_{\mathcal{D}\mathcal{D}} = \sigma_n^2\mathbf{I}$, $q^*(\mathbf{s}_\mathcal{I})$ (3.15) resembles that of DTC (see eqs. 68 and 69 in Appendix D.1.3 of [Hoang *et al.*, 2015]) except for the expectation over $\boldsymbol{\theta}$ due to the variational Bayesian treatment and coincides with that in Appendix B.1 of [Titsias and Lázaro-Gredilla, 2013]. So, we refer to it as VBDTC.

Remark 5. In the non-Bayesian setting of the hyperparameters, it has been previously established that the predictive distribution of FITC can be reproduced as a direct result of applying either variational inference [Titsias, 2009] with $\mathbf{C}_{\mathcal{D}\mathcal{D}} = \text{diag}[\mathbf{K}_{\mathcal{D}\mathcal{D}} - \mathbf{K}_{\mathcal{D}\mathcal{U}}\mathbf{K}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{K}_{\mathcal{U}\mathcal{D}}] + \sigma_n^2\mathbf{I}$ or expectation propagation (EP) [Bauer *et al.*, 2016] on the GP regression model. Our derivation of VBFITC is in fact similar in spirit to that of [Titsias, 2009] except for our variational Bayesian treatment of its hyperparameters. On the other hand, it is unclear whether FITC's equivalent EP derivation in [Bauer *et al.*, 2016] can be easily extended to incorporate a Bayesian treatment of its hyperparameters.

3.6 Stochastic Optimization

The VBDTC approximation [Titsias and Lázaro-Gredilla, 2013] has explicitly plugged the optimal $q^*(\mathbf{s}_\mathcal{I})$ (see Theorem 1) into $\mathcal{L}(q)$ (3.14) and reduced it to $\mathcal{L}(q)$ (B.4) in Appendix B.2. Given $\mathcal{L}(q)$ (B.4), the parameters $\boldsymbol{\nu}$ and $\boldsymbol{\Xi}$ of $q(\boldsymbol{\Lambda})$ and α and β of $q(\sigma_f)$ can be optimized via gradient ascent. However, evaluating the exact gradients $\partial\mathcal{L}/\partial\boldsymbol{\nu}$, $\partial\mathcal{L}/\partial\boldsymbol{\Xi}$, $\partial\mathcal{L}/\partial\alpha$ and $\partial\mathcal{L}/\partial\beta$ incur $\mathcal{O}(|\mathcal{D}||\mathcal{I}|^2)$ time, which scales poorly in the data size $|\mathcal{D}|$. To overcome the above issue of scalability, we utilize stochastic gradient ascent updates instead of exact ones, which requires the stochastic gradients to be unbiased estimators of the exact gradients to guarantee convergence. The key idea is to iteratively compute the stochastic gradients by randomly sampling

a single or few mini-batches of data from the dataset (i.e., comprising B disjoint mini-batches) whose incurred time per iteration is independent of data size $|\mathcal{D}|$. To achieve this, an important requirement is the decomposability of $\mathcal{L}(q)$ (B.4) into a summation of B terms, each of which is associated with a mini-batch $(\mathcal{D}_i, \mathbf{y}_{\mathcal{D}_i})$ of data of size $|\mathcal{D}_i| = \mathcal{O}(|\mathcal{I}|)$ that can be exploited for computing the stochastic gradients. Unfortunately, $\mathcal{L}(q)$ (B.4) is not decomposable due to its $(\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}} + \mathbf{\Psi}_{\mathcal{I}\mathcal{I}})^{-1}$ term. To remedy this, we do not plug $q^*(\mathbf{s}_\mathcal{I})$ (3.15) into $\mathcal{L}(q)$ (3.14) to yield (B.4) but instead jointly treat $q(\mathbf{s}_\mathcal{I})$, $q(\mathbf{\Lambda})$, and $q(\sigma_f)$ as variational parameters, which enables the decomposability of $\mathcal{L}(q)$ (3.14):

Corollary 1. $\mathcal{L}(q)$ (3.14) *(Theorem 1) can be decomposed into*

$$\begin{aligned}
\mathcal{L}(q) &= \sum_{i=1}^{B} \mathcal{L}_i(q) + \frac{1}{2}\Big(-\mathbf{m}^\top \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m} - \mathrm{Tr}[\mathbf{S}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}] + \log|\mathbf{S}| \\
&\qquad -\boldsymbol{\nu}^\top\boldsymbol{\nu} - \mathrm{Tr}[\mathbf{\Xi}] + \log|\mathbf{\Xi}| - \alpha^2 - \beta + \log\beta\Big) + \mathrm{const}\ , \\
\mathcal{L}_i(q) &\triangleq \frac{1}{2}\Big(2\mathbf{m}^\top \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Omega}_{\mathcal{I}\mathcal{D}_i}\mathbf{C}_{\mathcal{D}_i\mathcal{D}_i}^{-1}\mathbf{y}_{\mathcal{D}_i} - \mathbf{m}^\top \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Psi}_{\mathcal{I}\mathcal{I}}^{i}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m} \\
&\quad -\mathrm{Tr}[\mathbf{S}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Psi}_{\mathcal{I}\mathcal{I}}^{i}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}] - \mathrm{Tr}[\mathbf{C}_{\mathcal{D}_i\mathcal{D}_i}^{-1}\mathbf{\Upsilon}_{\mathcal{D}_i\mathcal{D}_i}] + \mathrm{Tr}[\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Psi}_{\mathcal{I}\mathcal{I}}^{i}]\Big)
\end{aligned}$$

where $\mathbf{\Psi}_{\mathcal{I}\mathcal{I}}^{i} \triangleq \mathbb{E}_{q(\boldsymbol{\theta})}[\mathbf{K}_{\mathcal{I}\mathcal{D}_i}\mathbf{C}_{\mathcal{D}_i\mathcal{D}_i}^{-1}\mathbf{K}_{\mathcal{D}_i\mathcal{I}}]$.

Our main result below exploits the decomposability of $\mathcal{L}(q)$ in Corollary 1 to derive stochastic gradients $\partial\widetilde{\mathcal{L}}/\partial\mathbf{m}$, $\partial\widetilde{\mathcal{L}}/\partial\mathbf{S}$, $\partial\widetilde{\mathcal{L}}/\partial\boldsymbol{\nu}$, $\partial\widetilde{\mathcal{L}}/\partial\mathbf{\Xi}$, $\partial\widetilde{\mathcal{L}}/\partial\alpha$, and $\partial\widetilde{\mathcal{L}}/\partial\beta$ that are unbiased estimators of their respective exact gradients, which is the key contribution of our work in this chapter:

Theorem 2. *Let $\mathcal{S}$ be a set of i.i.d. samples drawn from a uniform distribution over $\{1, 2, \ldots, B\}$. Construct the stochastic gradients $\partial\widetilde{\mathcal{L}}/\partial\mathbf{m}$, $\partial\widetilde{\mathcal{L}}/\partial\mathbf{S}$, $\partial\widetilde{\mathcal{L}}/\partial\boldsymbol{\nu}$, $\partial\widetilde{\mathcal{L}}/\partial\mathbf{\Xi}$, $\partial\widetilde{\mathcal{L}}/\partial\alpha$, and $\partial\widetilde{\mathcal{L}}/\partial\beta$ using the mini-batches $(\mathcal{D}_s, \mathbf{y}_{\mathcal{D}_s})$ for $s \in \mathcal{S}$ and current estimates of $(\mathbf{m}, \mathbf{S}, \boldsymbol{\nu}, \mathbf{\Xi}, \alpha, \beta)$ according to* (B.5) *in Appendix B.4.2. Then, $\mathbb{E}[\partial\widetilde{\mathcal{L}}/\partial\mathbf{m}] =$*

$\partial\mathcal{L}/\partial\mathbf{m}$, $\mathbb{E}[\partial\widetilde{\mathcal{L}}/\partial\mathbf{S}] = \partial\mathcal{L}/\partial\mathbf{S}$, $\mathbb{E}[\partial\widetilde{\mathcal{L}}/\partial\boldsymbol{\nu}] = \partial\mathcal{L}/\partial\boldsymbol{\nu}$, $\mathbb{E}[\partial\widetilde{\mathcal{L}}/\partial\boldsymbol{\Xi}] = \partial\mathcal{L}/\partial\boldsymbol{\Xi}$, $\mathbb{E}[\partial\widetilde{\mathcal{L}}/\partial\alpha] = \partial\mathcal{L}/\partial\alpha$, *and* $\mathbb{E}[\partial\widetilde{\mathcal{L}}/\partial\beta] = \partial\mathcal{L}/\partial\beta$.

Its proof is in Appendix B.3.

Remark 6. The stochastic gradients (Theorem 2) can be computed in closed form in $\mathcal{O}(|\mathcal{S}||\mathcal{I}|^3)$ time per iteration that reduces to $\mathcal{O}(|\mathcal{I}|^3)$ time by setting $|\mathcal{S}| = 1$ in our experiments. So, if the number of iterations of stochastic gradient ascent needed for convergence is much smaller than $t\min(|\mathcal{D}|/|\mathcal{I}|, B)$ where t is the required number of iterations of exact gradient ascent, then our stochastic variants achieve a huge speedup over the corresponding VBSGP regression models.

3.7 Bayesian Prediction with VBSGP Regression Models

Recall that the predictive distribution $p(f_{\mathbf{x}^*}|\mathbf{y}_\mathcal{D})$ is computationally intractable. We thus approximate it by

$$q(f_{\mathbf{x}^*}|\mathbf{y}_\mathcal{D}) = \int q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}_B}, \mathbf{s}_\mathcal{I}, \boldsymbol{\Lambda}, \sigma_f)\; q^+(\mathbf{s}_\mathcal{I})\; q^+(\boldsymbol{\Lambda})\; q^+(\sigma_f)\; \mathrm{d}\mathbf{s}_\mathcal{I}\; \mathrm{d}\boldsymbol{\Lambda}\; \mathrm{d}\sigma_\mathbf{f} \tag{3.16}$$

where $q^+(\mathbf{s}_\mathcal{I}) \triangleq \mathcal{N}(\mathbf{m}^+, \mathbf{S}^+)$, $q^+(\boldsymbol{\Lambda}) \triangleq \prod_{i=1}^d \mathcal{N}(\nu_i^+, \xi_i^+)$ with $\boldsymbol{\nu}^+ \triangleq (\nu_1^+, \ldots, \nu_d^+)^\top$ and $\boldsymbol{\Xi}^+ \triangleq \mathrm{diag}[\xi_1^+, \ldots, \xi_d^+]$, and $q(\sigma_f) \triangleq \mathcal{N}(\alpha^+, \beta^+)$ are obtained from the stochastic gradient ascent updates (Section 3.6). Note that $q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}_B}, \mathbf{s}_\mathcal{I}, \boldsymbol{\Lambda}, \sigma_f)$ is set to $p(f_{\mathbf{x}^*}|\mathbf{s}_\mathcal{I}, \boldsymbol{\Lambda}, \sigma_f)$ for the VBPITC, VBFIC, VBFITC, and VBDTC models and to $p(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}_B}, \mathbf{s}_\mathcal{I}, \boldsymbol{\Lambda}, \sigma_f)$ for the VBPIC model. Although the predictive distribution $q(f_{\mathbf{x}^*}|\mathbf{y}_\mathcal{D})$ is not Gaussian, its predictive mean $\mu_{\mathbf{x}^*|\mathcal{D}}$ and variance $\sigma^2_{\mathbf{x}^*|\mathcal{D}}$ can be computed analytically for VBPITC, VBFIC, VBFITC, and VBDTC and via sampling for VBPIC, as derived in Appendix B.5. Their respective predictive means $\mu_{\mathbf{x}^*|\mathcal{D}}$ closely

resemble that of PITC, FIC, FITC, DTC, and PIC (see eqs. 84 and 86 in Appendix D.4 of [Hoang *et al.*, 2015]) except for the expectations over $\mathbf{\Lambda}$ and σ_f due to the variational Bayesian treatment. Their predictive variances $\sigma^2_{\mathbf{x}^*|\mathcal{D}}$ are also similar except for the expectations over $\mathbf{\Lambda}$ and σ_f and an additional positive term arising from the uncertainty of $\mathbf{\Lambda}$ and σ_f.

3.8 Experiments and Discussion

This section empirically evaluates the predictive performance and time efficiency of the stochastic variants, denoted by VBDTC+, VBFITC+, and VBPIC+, of our VBSGP regression models (respectively, VBDTC, VBFITC, and VBPIC). We will first use the small AIMPEAK dataset [Chen *et al.*, 2013] on traffic speeds of size 41850 to evaluate the convergence of the variational distributions $q^+(\mathbf{s}_\mathcal{I})$ and $q^+(\mathbf{\Lambda}, \sigma_f)$ induced by our stochastic variants VBDTC+, VBFITC+, and VBPIC+ to, respectively, $q(\mathbf{s}_\mathcal{I})$ and $q(\mathbf{\Lambda}, \sigma_f)$ induced by VBDTC [Titsias and Lázaro-Gredilla, 2013], VBFITC, and VBPIC performing exact gradient ascent updates via *scaled conjugate gradient* (SCG). To do this, we use the KL distance metric to measure the distance between the variational distributions obtained from the stochastic vs. exact gradient ascent.

Then, using the real-world TWITTER dataset on buzz events of size 583250 and AIRLINE dataset [Hensman *et al.*, 2013] on flight delays of size 2055733, we will compare the performance of the stochastic variants of our VBSGPR models with that of the state-of-the-art GP models such as the stochastic variants of variational DTC (SVIGP) [Hensman *et al.*, 2013] and variational PIC (PIC+) [Hoang *et al.*, 2015], distributed variational DTC (Dist-VGP) [Gal *et al.*, 2014], and rBCM [Deisenroth and Ng, 2015], all of which find point estimates of hyperparameters. Such a comparison will demonstrate the benefits of adopting a variational Bayesian treatment of the hyperparameters by our VBSGP regression models. We will also compare the

performance of our stochastic VBSGP regression models with that of the stochastic variant of *variational Bayesian sparse spectrum GP* (VSSGP) regression model [Gal and Turner, 2015]. To evaluate their predictive performance, we use the *root mean square error* (RMSE) metric: $\sqrt{\sum_{\mathbf{x}^*\in\mathcal{T}}(y_{\mathbf{x}^*}-\mu_{\mathbf{x}^*|\mathcal{D}})^2/|\mathcal{T}|}$ and the *mean negative log probability* (MNLP) metric: $0.5\sum_{\mathbf{x}^*\in\mathcal{T}}\{(y_{\mathbf{x}^*}-\mu_{\mathbf{x}^*|\mathcal{D}})^2/\sigma^2_{x^*|\mathcal{D}}+\log(2\pi\sigma^2_{x^*|\mathcal{D}})\}/|\mathcal{T}|$ where $\mathcal{T}$ denotes a set of test inputs.

All datasets are modeled using GPs whose prior covariance is defined by the squared exponential covariance function defined in Section 3.2. All experiments are run on a Linux system with Intel® Xeon® E5-2683 CPU at 2.1GHz with 256GB memory.

3.8.1 Empirical Convergence of Stochastic VBSGP Regression Models

The AIMPEAK dataset [Chen *et al.*, 2013] of size 41850 comprises traffic speeds (km/h) along 775 road segments of an urban road network during morning peak hours on April 20, 2011. Each input (i.e., road segment) denotes a 5D feature vector of length, number of lanes, speed limit, direction, and time, the last of which comprises 54 five-minute time slots. The output corresponds to traffic speed. We randomly select training data of size 1000, which is partitioned into $B = 10$ mini-batches, and 50 inducing inputs from the inputs of the training data.

Figs. 3.1a-3.1c (Figs. 3.1d-3.1f) shows results of the KL distance $\mathrm{KL}(q(\mathbf{s}_\mathcal{I})||q^+(\mathbf{s}_\mathcal{I}))$ ($\mathrm{KL}(q(\mathbf{\Lambda},\sigma_f)||q^+(\mathbf{\Lambda},\sigma_f))$) of $q^+(\mathbf{s}_\mathcal{I})$ to $q(\mathbf{s}_\mathcal{I})$ ($q^+(\mathbf{\Lambda},\sigma_f)$ to $q(\mathbf{\Lambda},\sigma_f)$) averaged over 5 random selections of training data and mini-batch sequences with an increasing number t of iterations. It can be observed that the variational distributions $q^+(\mathbf{s}_\mathcal{I})$ and $q^+(\mathbf{\Lambda},\sigma_f)$ induced by VBDTC+, VBFITC+, and VBPIC+ converge rapidly to, respectively, $q(\mathbf{s}_\mathcal{I})$ and $q(\mathbf{\Lambda},\sigma_f)$ induced by VBDTC, VBFITC, and VBPIC, thus

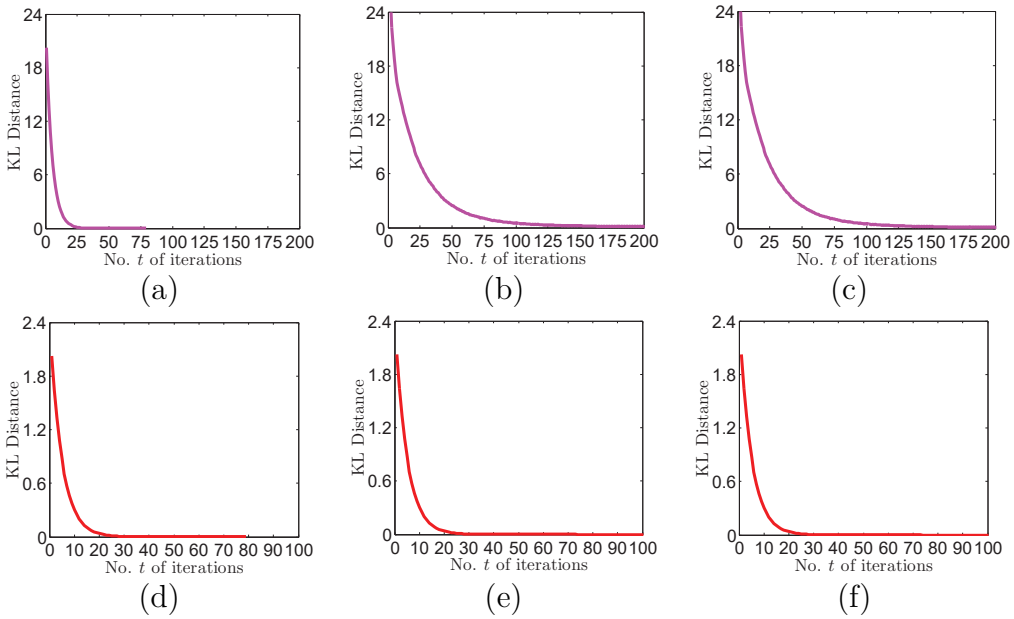


Figure 3.1: Graphs of KL distance $\mathrm{KL}(q(\mathbf{s}_\mathcal{I})||q^+(\mathbf{s}_\mathcal{I}))$ of (a) VBDTC+ to VBDTC, (b) VBFITC+ to VBFITC, (c) VBPIC+ to VBPIC, and $\mathrm{KL}((q(\mathbf{\Lambda},\sigma_f)||q^+(\mathbf{\Lambda},\sigma_f))$ of (d) VBDTC+ to VBDTC, (e) VBFITC+ to VBFITC, (f) VBPIC+ to VBPIC vs. no. t of iterations for AIMPEAK dataset.

corroborating our theoretical results in Section 3.6. From Figs. 3.1a-3.1c, it can also be observed that $q^+(\mathbf{s}_\mathcal{I})$ induced by VBDTC+ converges faster to $q(\mathbf{s}_\mathcal{I})$ than that by VBFITC+ and VBPIC+, which can be explained by its much simpler noise structure by assuming i.i.d. observation noises with constant variance σ_n^2.

3.8.2 Empirical Evaluation on AIRLINE and TWITTER Datasets

The TWITTER dataset contains 583250 instances of buzz events on Twitter. The input denotes a relatively large 77D feature vector[2], which makes this dataset suitable for evaluating robustness to overfitting. The output is the popularity of the instance's topic. The massive benchmark AIRLINE dataset [Hensman *et al.*, 2013] contains 2055733 records of information about every commercial flight in the USA

[2]Detailed description is at http://ama.liglab.fr/datasets/buzz/

from January to April 2008. The input denotes an 8D feature vector of age of the aircraft (no. of years in service), travel distance (km), airtime, departure and arrival time (min.) as well as day of the week, day of month, and month. The output is the delay time (min.) of the flight. For each dataset, 5% is randomly selected and set aside as test data. The remaining data is used as training data and partitioned into $B = 1000$ mini-batches using k-means (i.e., $k = B$). We randomly select 100 inducing inputs from the inputs of the training data.

Figs. 3.2a and 3.2b show results of RMSE and MNLP achieved by the stochastic variants of our VBSGPR models averaged over 5 random selections of 5% test data and mini-batch sequences with an increasing number t of iterations for the AIRLINE dataset. It can be observed that VBPIC+ (RMSE of 21.87 min. and MNLP of 4.53) achieves considerably better predictive performance than VBFITC+ (RMSE of 37.05 min. and MNLP of 7.84) and VBDTC+ (RMSE of 37.55 min. and MNLP of 8.06). To explain this, VBFITC+ and VBDTC+ have both imposed a strong assumption of independently distributed observation noises. In contrast, VBPIC+ caters to correlation of observation noises within each mini-batch of data (Sections 3.5 and 3.6), hence modeling and predicting real-world datasets with correlated noises better. Furthermore, unlike VBFITC+ and VBDTC+, VBPIC+ does not assume conditional independence between the training and test outputs given the inducing outputs in its test conditional.

Fig. 3.2c exhibits a near-linear increase in total incurred time with an increasing number t of iterations for VBDTC+, VBFITC+, and VBPIC+. Our experiments reveal that VBDTC+, VBFITC+, and VBPIC+ incur, respectively, an average of 0.0122, 0.0132, and 0.038 seconds per iteration of stochastic gradient ascent update. Fig. 3.2d shows that VBPIC+ can achieve a more superior trade-off between predictive performance vs. time efficiency than VBDTC+ and VBFITC+.

Figs. 3.3a and 3.3b show results of RMSE and MNLP achieved by the stochastic

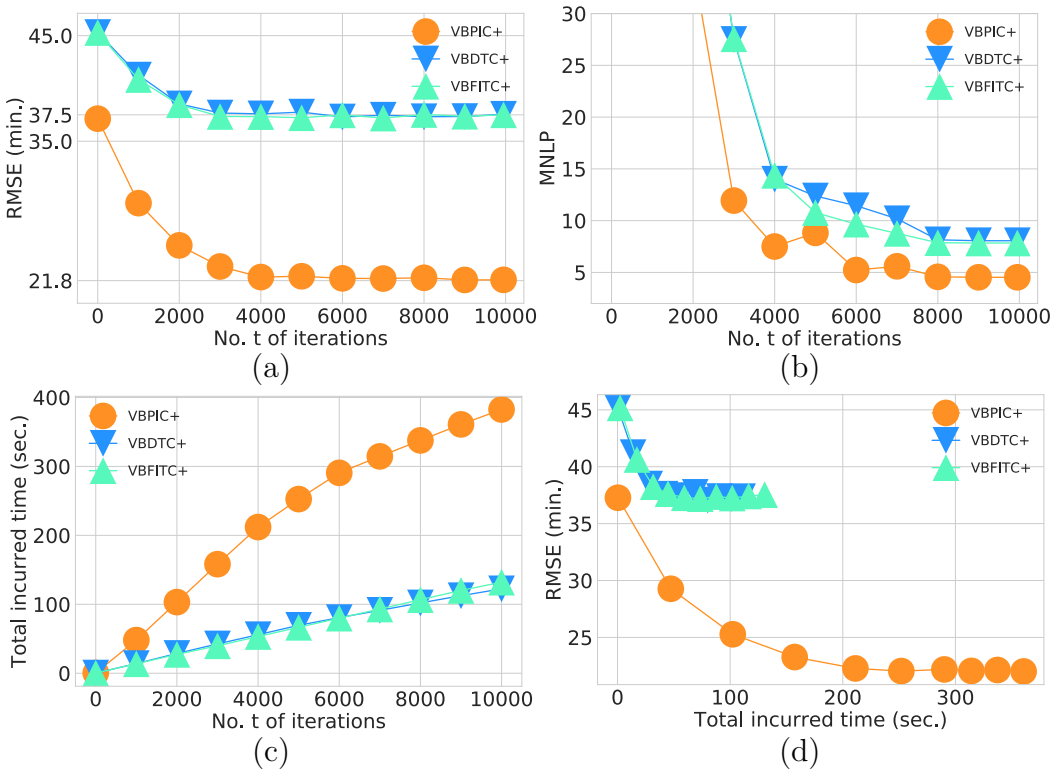


Figure 3.2: Graphs of (a) RMSE, (b) MNLP, and (c) total incurred time vs. number t of iterations, and (d) graphs of RMSE vs. total incurred time of VBDTC+, VBFITC+, and VBPIC+ for the AIRLINE dataset.

variants of our VBSGP regression models averaged over 5 random selections of 5% test data and mini-batch sequences with an increasing number t of iterations for the TWITTER dataset. The observations are similar to that for the AIRLINE dataset: It can be observed that VBPIC+ (RMSE of 131.46 and MNLP of 6.45) achieves significantly better predictive performance than VBFITC+ (RMSE of 212.67 and MNLP of 7.21) and VBDTC+ (RMSE of 247.38 and MNLP of 7.69); this can be explained by the same reasons as that discussed previously for the AIRLINE dataset.

Fig. 3.3c also exhibits a linear increase in total incurred time with an increasing number t of iterations for VBDTC+, VBFITC+, and VBPIC+. Our experiments

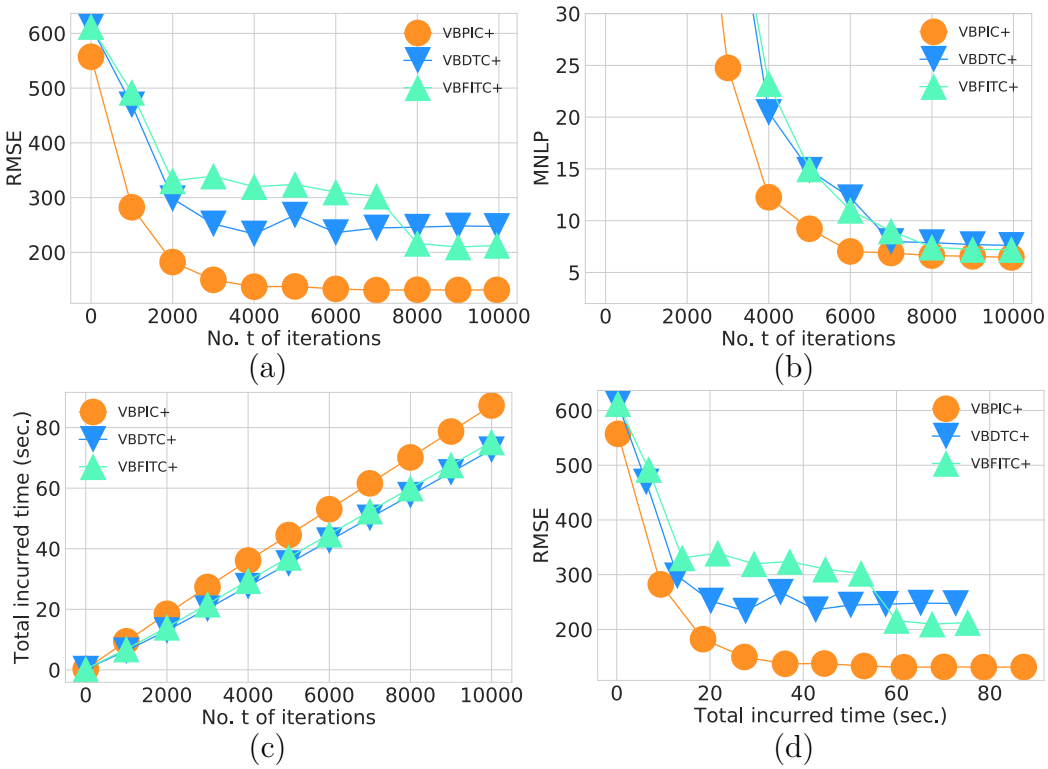


Figure 3.3: Graphs of (a) RMSE, (b) MNLP, and (c) total incurred time vs. number t of iterations, and (d) graphs of RMSE vs. total incurred time of VBDTC+, VBFITC+, and VBPIC+ for the TWITTER dataset.

reveal that VBDTC+, VBFITC+, and VBPIC+ incur, respectively, an average of 0.0073, 0.0075, and 0.0087 seconds per iteration of stochastic gradient ascent update, which are shorter than that for the AIRLINE dataset due to a smaller mini-batch size. Fig. 3.3d reveals that VBPIC+ can similarly achieve the best trade-off between predictive performance vs. time efficiency.

Table 3.1 compares the predictive performance (RMSEs) achieved by state-of-the-art GP models for the AIRLINE and TWITTER datasets. It can be observed that our VBPIC+ significantly outperforms state-of-the-art SVIGP, Dist-VGP, rBCM, and PIC+, which find point estimates of hyperparameters, and VSSGP due to its restrictive assumption, as discussed in Section 3.1. In contrast, our VBPIC+ assumes

Dataset	SVIGP	Dist-VGP	rBCM	PIC+	VBPIC+	VSSGPR
AIRLINE	39.53	35.30	34.40	24.9	**21.87**	38.95
TWITTER	–	–	–	190.2	**131.4**	585.9

Table 3.1: RMSE achieved by VBPIC+ and state-of-the-art GP models for AIRLINE and TWITTER datasets. The results of PIC+ and VSSGPR are obtained using their GitHub codes. The results of Dist-VGP and rBCM are taken from their respective papers and that of SVIGP is reported in [Hoang *et al.*, 2015]. They are all based on the same settings of training/test data sizes = 2M/100K (554K/29K) for the AIRLINE (TWITTER) dataset.

a variational Bayesian treatment of its hyperparameters, thus achieving robustness to overfitting due to Bayesian model selection, as demonstrated later. Unlike VSSGP, VBPIC+ does not assume conditional independence between the training and test outputs in its test conditional.

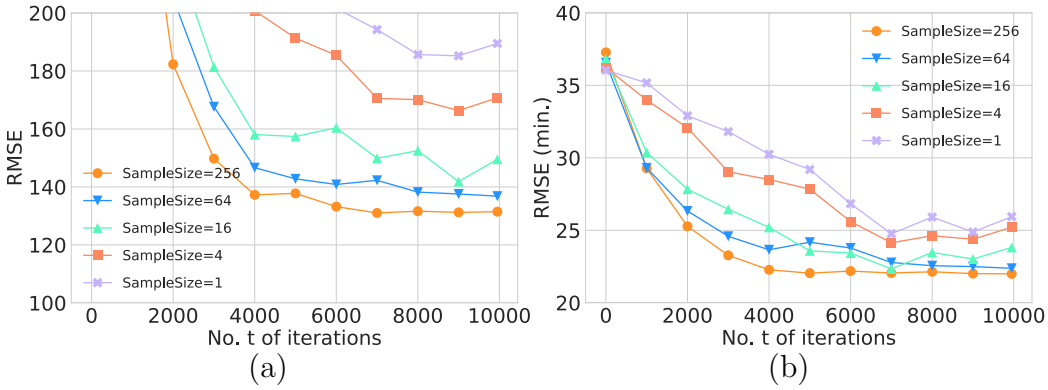


Figure 3.4: Graphs of RMSEs of VBPIC+ vs. number t of iterations with varying sampling sizes for computing its predictive mean for the (a) TWITTER and (b) AIRLINE datasets.

Fig. 3.4 shows results of RMSEs achieved by our VBPIC+ with an increasing number t of iterations and varying sample sizes for computing its predictive mean (Section 3.7). Note that a sample size of 1 reduces VBPIC to PIC that treats its sampled hyperparameters as a point estimate. By increasing the sample size, it can be observed that VBPIC+ converges faster to a lower RMSE using less iterations due to its Bayesian model selection/averaging, thus demonstrating its increasing robustness to overfitting.

Fig. 3.5 displays the 95% confidence intervals (mean $\nu_i^+ \pm 2 \times$ standard deviation $(\xi_i^+)^{1/2}$) for inverted length-scale hyperparameters λ_i for $i = 1, \ldots, d$ after $t = 10000$ iterations for the TWITTER ($d = 77$ normalized input dimensions) and AIRLINE ($d = 8$ normalized input dimensions) datasets. It can be observed that the confidence intervals are generally wider (i.e., larger uncertainty of $\lambda_1, \ldots, \lambda_d$) for the TWITTER dataset than for the AIRLINE dataset. To confirm this, we measure the *mean log variance* (MLV) $\sum_{i=1}^{d} \log \xi_i^+ / d$ of $\lambda_1, \ldots, \lambda_d$ and notice that the TWITTER dataset gives a higher MLV of -4.09 than that for the AIRLINE dataset (i.e., MLV $= -6.55$). So, with a larger uncertainty of $\lambda_1, \ldots, \lambda_d$, their point estimates have a greater tendency to overfit and hence yield a poorer predictive performance, as observed in Fig. 3.4 (compare the performance gap between sample sizes of 1 vs. 256).

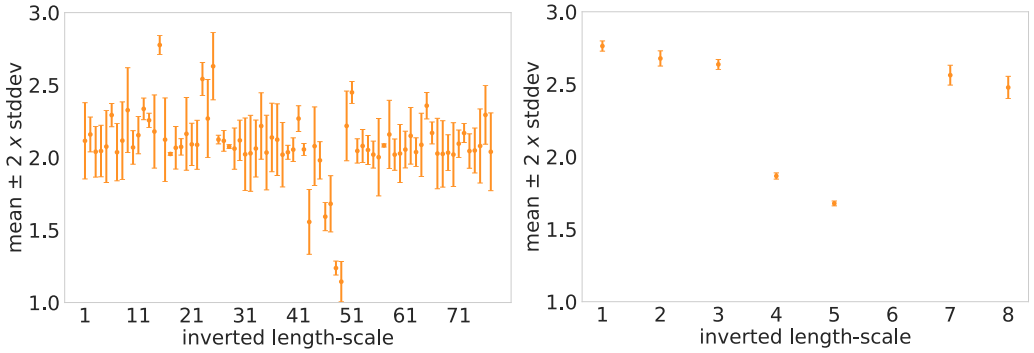


Figure 3.5: 95% confidence intervals (mean $\nu_i^+ \pm 2 \times$ standard deviation $(\xi_i^+)^{1/2}$) for inverted length-scale hyperparameters λ_i for $i = 1, \ldots, d$ after $t = 10000$ iterations for the (a) TWITTER ($d = 77$ normalized input dimensions) and (b) AIRLINE ($d = 8$ normalized input dimensions) datasets.

3.9 Summary

This chapter describes a novel VI framework for a family of VBSGP regression models (e.g., VBDTC, VBFITC, VBPIC) whose approximations are variationally optimal

with respect to the GP regression model enriched with various corresponding correlation structures of the observation noises. Our variational Bayesian treatment of hyperparameters enables our VBSGP regression models to mitigate critical issues (e.g., overfitting) which plague existing variational SGP regression models that optimize point estimates of hyperparameters. The stochastic variants of our VBSGP regression models can yield good predictive performance fast and improve their predictive performance over time, thus achieving scalability to big data. Empirical evaluation on two real-world datasets reveals that the stochastic variant of our VBPIC can significantly outperform existing state-of-the-art GP models, thus demonstrating its robustness to overfitting due to Bayesian model selection while preserving scalability to big data through stochastic optimization.

Chapter 4

Implicit Posterior Variational Inference for Deep Gaussian Processes

This chapter is based on the paper published in the 2019 Neural Information Processing Systems (NeurIPS-19): "Implicit Posterior Variational Inference for Deep Gaussian Process".

4.1 Introduction

Chapter 3 has proposed a family of VBSGP regression models which mitigate overfitting issues as well as outperform existing state-of-the-art GP regression models. Recall from Section 1.1.4 that GP models are fully specified by its covariance/kernel function[1]. hence, the capability of GP models is limited by the expressiveness of the kernel function, which makes them difficult in modeling real-world complex datasets.

[1]Commonly, the mean function is assumed to be zero.

Moreover, it is challenging to design an appropriate kernel function without having prior knowledge about the underlying properties and structures of the data.

To this end, a multi-layer hierarchical composition of GPs (DGP) [Damianou and Lawrence, 2013] is proposed to overcome the limitations of single-layer GPs while retaining the advantages. It has been proved by Neal (1995) that a single-layer GP is equivalent to an infinitely wide neural network with a single hidden layer. Similarly, it can be shown that a DGP is equivalent to a multi-layer neural network with multiple infinitely wide hidden layers. The mapping between layers in this type of neural network is parametrized by a GP, and, as a result, DGPs retain useful theoretical properties of GPs such as nonparametric modeling power and well-calibrated predictive uncertainty estimate. In addition, DGPs are arguably more flexible, have a greater capacity to generalize, and are potentially able to provide better predictive performance [Damianou and Lawrence, 2013; Salimbeni and Deisenroth, 2017; Havasi *et al.*, 2018]. DGPs are richer models than single-layer GPs, which is analogous to deep neural networks being richer than generalized linear models.

Similarly, the posterior for DGP does not entail tractable inference. This has also motivated the development of deterministic and stochastic approximation methods. Particularly, previous literatures on deterministic approximation are based on variational inference [Damianou and Lawrence, 2013; Hensman and Lawrence, 2014; Dai *et al.*, 2016; Salimbeni and Deisenroth, 2017] and expectation propagation [Bui *et al.*, 2016]. This unifying framework have imposed varying structural assumptions across the DGP hidden layers and a Gaussian posterior belief of the inducing variables. However, the work of [Havasi *et al.*, 2018] has demonstrated that the posterior belief of the inducing variables is usually non-Gaussian, hence potentially compromising the performance of the deterministic approximation methods due to the biased posterior belief. To resolve this, Havasi *et al.* utilizes *stochastic gradient Hamiltonian Monte Carlo* (SGHMC) sampling to draw unbiased samples from the posterior

belief. However, generating such samples is computationally costly in both training and prediction due to its sequential sampling procedure [Wang *et al.*, 2018] and its convergence is also difficult to assess. Therefore, the challenge remains in devising a time-efficient approximation method that can recover an unbiased posterior belief for inference in DGP.

The focus of this chapter is to present an *implicit posterior variational inference* framework for DGPs (Section 4.3) that can ideally recover an unbiased posterior belief and still preserve time efficiency, hence combining the best of both worlds (respectively, stochastic and deterministic approximation methods). Inspired by generative adversarial networks [Goodfellow *et al.*, 2014] that can generate samples to represent complex distributions which are hard to model using an explicit likelihood [Karras *et al.*, 2018; van den Oord *et al.*, 2016], our IPVI framework achieves this by casting the DGP inference problem as a two-player game in which a Nash equilibrium, interestingly, coincides with an unbiased posterior belief. We empirically evaluate the performance of IPVI on several real-world datasets in supervised (e.g., regression and classification) and unsupervised learning tasks (Section 4.4).

4.2 Deep Gaussian Processes

In this section, we will review the technical details for Deep Gaussian Processes (DGP), suppose that a set $\mathbf{y} \triangleq \{y_n\}_{n=1}^N$ of N noisy observed outputs are available for some set $\mathbf{X} \triangleq \{\mathbf{x}_n\}_{n=1}^N$ of N training inputs. A multi-layer DGP model is a hierarchical composition of GP models.

Consider a DGP with a depth of L such that each DGP layer is associated with a set $\mathbf{F}_{\ell-1}$ of inputs and a set $\mathbf{F}_\ell$ of outputs for $\ell = 1, \ldots, L$ and $\mathbf{F}_0 \triangleq \mathbf{X}$. Let $\boldsymbol{\mathcal{F}} \triangleq \{\mathbf{F}_\ell\}_{\ell=1}^L$, and the inducing inputs and corresponding inducing output variables for DGP layers $\ell = 1, \ldots, L$ be denoted by the respective sets $\boldsymbol{\mathcal{Z}} \triangleq \{\mathbf{Z}_\ell\}_{\ell=1}^L$ and

$\boldsymbol{\mathcal{U}} \triangleq \{\mathbf{U}_\ell\}_{\ell=1}^L$. Similar to the joint probability distribution of the SGP model in Chapter 3,

$$p(\mathbf{y}, \boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{U}}) = \underbrace{p(\mathbf{y}|\mathbf{F}_L)}_{\text{data likelihood}} \underbrace{\left[\prod_{\ell=1}^{L} p(\mathbf{F}_\ell|\mathbf{U}_\ell)\right] p(\boldsymbol{\mathcal{U}})}_{\text{DGP prior}}.$$

Similarly, the variational posterior is assumed to be $q(\boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{U}}) \triangleq \left[\prod_{\ell=1}^{L} p(\mathbf{F}_\ell|\mathbf{U}_\ell)\right] q(\boldsymbol{\mathcal{U}})$, thus resulting in the following ELBO for the DGP model:

$$\text{ELBO} \triangleq \int q(\mathbf{F}_L) \log p(\mathbf{y}|\mathbf{F}_L) \, \mathrm{d}\mathbf{F}_L - \mathrm{KL}[q(\boldsymbol{\mathcal{U}}) \| p(\boldsymbol{\mathcal{U}})] \tag{4.1}$$

where $q(\mathbf{F}_L) \triangleq \int \prod_{\ell=1}^{L} p(\mathbf{F}_\ell|\mathbf{U}_\ell, \mathbf{F}_{\ell-1}) \; q(\boldsymbol{\mathcal{U}}) \; \mathrm{d}\mathbf{F}_1 \ldots \mathrm{d}\mathbf{F}_{L-1} \mathrm{d}\boldsymbol{\mathcal{U}}$. To compute $q(\mathbf{F}_L)$, the work of [Salimbeni and Deisenroth, 2017] has proposed the use of the reparameterization trick [Kingma and Welling, 2013] and Monte Carlo sampling, which are adopted in this chapter.

Remark 7. To the best of our knowledge, previous works on DGP models exploiting the inducing variables[2] and the VI framework [Dai *et al.*, 2016; Damianou and Lawrence, 2013; Hensman and Lawrence, 2014; Salimbeni and Deisenroth, 2017] have imposed the following highly restrictive assumptions

- mean field approximation $q(\boldsymbol{\mathcal{U}}) \triangleq \prod_{\ell=1}^{L} q(\mathbf{U}_\ell)$.
- biased Gaussian variational posterior $q(\mathbf{U}_\ell)$.

In fact, the true posterior belief usually exhibits a high correlation across the DGP layers and is non-Gaussian [Havasi *et al.*, 2018], hence potentially compromising the performance of such deterministic approximation methods for DGP models. To remove these assumptions, we will propose a principled approximation method that

[2] An alternative is to modify the DGP prior directly and perform inference with a parametric model. The work of [Cutajar *et al.*, 2017] has approximated the DGP prior with the spectral density of a kernel such that the kernel has an analytical spectral density.

can generate unbiased posterior samples even under the VI framework, as detailed in Section 4.3.

4.3 Implicit Posterior Variational Inference (IPVI)

Unlike the existing VI framework for DGP models [Dai *et al.*, 2016; Damianou and Lawrence, 2013; Hensman and Lawrence, 2014; Salimbeni and Deisenroth, 2017], our proposed IPVI framework does not need to impose their highly restrictive assumptions (Remark 7) and can still preserve the time efficiency of VI. Inspired by previous works on adversarial-based inference [Huszár, 2017; Mescheder *et al.*, 2017], IPVI achieves this by first generating posterior samples $\boldsymbol{\mathcal{U}} \triangleq g_\Phi(\epsilon)$ with a black-box **generator** $g_\Phi(\epsilon)$ parameterized by Φ and a random noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. By representing the variational posterior as $q_\Phi(\boldsymbol{\mathcal{U}}) \triangleq \int p(\boldsymbol{\mathcal{U}}|\epsilon)\mathrm{d}\epsilon$, the ELBO in (4.1) can be re-written as

$$\text{ELBO} = \mathbb{E}_{q(\mathbf{F}_L)}[\log p(\mathbf{y}|\mathbf{F}_L)] - \text{KL}[q_\Phi(\boldsymbol{\mathcal{U}})\|p(\boldsymbol{\mathcal{U}})] \ . \tag{4.2}$$

An immediate advantage of the generator $g_\Phi(\epsilon)$ is that it can generate the posterior samples in parallel by feeding it a batch of randomly sampled ϵ's. However, representing the variational posterior $q_\Phi(\boldsymbol{\mathcal{U}})$ implicitly makes it impossible to evaluate the KL distance in (4.2) since $q_\Phi(\boldsymbol{\mathcal{U}})$ cannot be calculated explicitly. By observing that the KL distance is equal to the expectation of the log-density ratio $\mathbb{E}_{q_\Phi(\boldsymbol{\mathcal{U}})}[\log q_\Phi(\boldsymbol{\mathcal{U}}) - \log p(\boldsymbol{\mathcal{U}})]$, we can circumvent an explicit calculation of the KL distance term by implicitly representing the log-density ratio as a separate function T to be optimized, as shown in our first result below:

Proposition 1. *Let $\sigma(x) \triangleq 1/(1+\exp(-x))$. Consider the following maximization problem:*

$$\max_T \ \mathbb{E}_{p(\boldsymbol{\mathcal{U}})}[\log(1-\sigma(T(\boldsymbol{\mathcal{U}}))] + \mathbb{E}_{q_\Phi(\boldsymbol{\mathcal{U}})}[\log\sigma(T(\boldsymbol{\mathcal{U}}))] \ . \tag{4.3}$$

If $p(\boldsymbol{\mathcal{U}})$ and $q_\Phi(\boldsymbol{\mathcal{U}})$ are known, then the optimal T^* with respect to (4.3) *is the log-density ratio:*

$$T^*(\boldsymbol{\mathcal{U}}) = \log q_\Phi(\boldsymbol{\mathcal{U}}) - \log p(\boldsymbol{\mathcal{U}}) \ . \tag{4.4}$$

Its proof (Appendix C.1) is similar to that of Proposition 1 in [Goodfellow *et al.*, 2014] except that we use a sigmoid function σ to reveal the log-density ratio. Note that (4.3) defines a binary cross-entropy between samples from the variational posterior $q_\Phi(\boldsymbol{\mathcal{U}})$ and prior $p(\boldsymbol{\mathcal{U}})$. Intuitively, T in (4.3), which we refer to as a **discriminator**, tries to distinguish between $q_\Phi(\boldsymbol{\mathcal{U}})$ and $p(\boldsymbol{\mathcal{U}})$ by outputting $\sigma(T(\boldsymbol{\mathcal{U}}))$ as the probability of $\boldsymbol{\mathcal{U}}$ being a sample from $q_\Phi(\boldsymbol{\mathcal{U}})$ rather than $p(\boldsymbol{\mathcal{U}})$.

Using Proposition 1 (i.e., (4.4)), the ELBO in (4.2) can be re-written as

$$\text{ELBO} = \mathbb{E}_{q_\Phi(\boldsymbol{\mathcal{U}})}[\mathcal{L}(\theta, \mathbf{X}, \mathbf{y}, \boldsymbol{\mathcal{U}}) - T^*(\boldsymbol{\mathcal{U}})] \tag{4.5}$$

where $\mathcal{L}(\theta, \mathbf{X}, \mathbf{y}, \boldsymbol{\mathcal{U}}) \triangleq \mathbb{E}_{p(\mathbf{F}_L|\boldsymbol{\mathcal{U}})}[\log p(\mathbf{y}|\mathbf{F}_L)]$ and θ denotes the DGP model hyperparameters. The ELBO can now be calculated given the optimal discriminator T^*. In our implementation, we adopt a parametric representation for discriminator T. In principle, the parametric representation is required to be expressive enough to be able to represent the optimal discriminator T^* accurately. Motivated by the fact that deep neural networks are universal function approximators [Hornik *et al.*, 1989], we represent discriminator T_Ψ by a neural network with parameters Ψ; the optimal T_{Ψ^*} is thus parameterized by Ψ^*.

The ELBO in (4.5) can be optimized with respect to Φ and θ via gradient ascent, provided that the optimal T_{Ψ^*} (with respect to q_Φ) can be obtained in every iteration. One way to achieve this is to cast the optimization of the ELBO as a two-player pure-strategy game between **Player 1** (representing discriminator with strategy $\{\Psi\}$) vs. **Player 2** (jointly representing generator and DGP model with strategy $\{\Phi, \theta\}$) that

is defined based on the following payoffs:

$$\begin{aligned} &\textbf{Player 1}: \max_{\{\Psi\}} \ \mathbb{E}_{p(\boldsymbol{\mathcal{U}})}[\log(1-\sigma(T_\Psi(\boldsymbol{\mathcal{U}}))] + \mathbb{E}_{q_\Phi(\boldsymbol{\mathcal{U}})}[\log\sigma(T_\Psi(\boldsymbol{\mathcal{U}}))] \ , \\ &\textbf{Player 2}: \max_{\{\theta,\Phi\}} \ \mathbb{E}_{q_\Phi(\boldsymbol{\mathcal{U}})}[\mathcal{L}(\theta,\mathbf{X},\mathbf{y},\boldsymbol{\mathcal{U}}) - T_\Psi(\boldsymbol{\mathcal{U}})] \ . \end{aligned} \tag{4.6}$$

Proposition 2. *Suppose that the parametric representations of T_Ψ and g_Φ are expressive enough to represent any function. If $(\{\Psi^*\},\{\theta^*,\Phi^*\})$ is a Nash equilibrium of the game in* (4.6)*, then $\{\theta^*,\Phi^*\}$ is a global maximizer of the ELBO in* (4.2) *such that (a) θ^* is the maximum likelihood assignment for the DGP model, and (b) $q_{\Phi^*}(\boldsymbol{\mathcal{U}})$ is equal to the true posterior belief $p(\boldsymbol{\mathcal{U}}|\mathbf{y})$.*

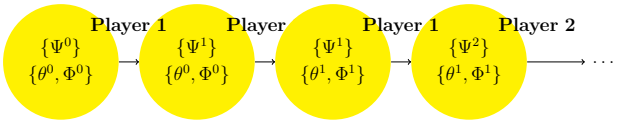
Its proof is similar to that of Proposition 3 in [Mescheder *et al.*, 2017] except that we additionally provide a proof of existence of a Nash equilibrium for the case of known/fixed DGP model hyperparameters, as detailed in Appendix C.2. Proposition 2 reveals that any Nash equilibrium coincides with a global maximizer of the original ELBO in (4.2). This consequently inspires us to play the game using *best-response dynamics*[3] (BRD) which is a commonly adopted procedure [Awerbuch *et al.*, 2008] to search for a Nash equilibrium. Fig. 4.1 illustrates our BRD algorithm: In each iteration of Algorithm 1, each player takes its turn to improve its strategy to achieve a better (but not necessarily the best) payoff by performing a *stochastic gradient ascent* (SGA) update on its payoff (4.6).

Remark 8. While BRD guarantees to converge to a Nash equilibrium in some classes of games (e.g., a finite potential game), we have not shown that our game falls into any of these classes and hence cannot guarantee that BRD converges to a Nash equilibrium (i.e., global maximizer $\{\theta^*,\Phi^*\}$) of our game. Nevertheless, as mentioned previously, obtaining the optimal discriminator in every iteration guarantees the game

[3]This procedure is sometimes called "better-response dynamics" (`http://timroughgarden.org/f13/l/l16.pdf`).

Algorithm 1: Main

Randomly initialize θ, Ψ, Φ
while *not converged* **do**
  Run Algorithm 2
  Run Algorithm 3



Algorithm 2: Player 1

Sample $\{\boldsymbol{\mathcal{V}}_1, \ldots, \boldsymbol{\mathcal{V}}_K\}$ from $p(\boldsymbol{\mathcal{U}})$
Sample $\{\boldsymbol{\mathcal{U}}_1, \ldots, \boldsymbol{\mathcal{U}}_K\}$ from $q_\Phi(\boldsymbol{\mathcal{U}})$
Compute gradient w.r.t. Ψ from (4.6):

$$g_\Psi \triangleq \nabla_\Psi \left[\frac{1}{K} \sum_{k=1}^{K} \log(1 - \sigma(T_\Psi(\boldsymbol{\mathcal{V}}_k)) \right] + \nabla_\Psi \left[\frac{1}{K} \sum_{k=1}^{K} \log \sigma(T_\Psi(\boldsymbol{\mathcal{U}}_k)) \right]$$

SGA update for Ψ:

$$\Psi \leftarrow \Psi + \alpha_\Psi \; g_\Psi$$

Algorithm 3: Player 2

Sample mini-batch $(\mathbf{X}_b, \mathbf{y}_b)$ from $(\mathbf{X}, \mathbf{y})$
Sample $\{\boldsymbol{\mathcal{U}}_1, \ldots, \boldsymbol{\mathcal{U}}_K\}$ from $q_\Phi(\boldsymbol{\mathcal{U}})$
Compute gradients w.r.t. θ and Φ from (4.6):

$$g_\theta \triangleq \nabla_\theta \left[\frac{1}{K} \sum_{k=1}^{K} \mathcal{L}(\theta, \mathbf{X}_b, \mathbf{y}_b, \boldsymbol{\mathcal{U}}_k) \right]$$

$$g_\Phi \triangleq \nabla_\Phi \left[\frac{1}{K} \sum_{k=1}^{K} \mathcal{L}(\theta, \mathbf{X}_b, \mathbf{y}_b, \boldsymbol{\mathcal{U}}_k) - T_\Psi(\boldsymbol{\mathcal{U}}_k) \right]$$

SGA updates for θ and Φ:

$$\theta \leftarrow \theta + \alpha_\theta \; g_\theta \;, \quad \Phi \leftarrow \Phi + \alpha_\Phi \; g_\Phi$$

Figure 4.1: *Best-response dynamics* (BRD) algorithm based on our IPVI framework for DGPs.

play (i.e., gradient ascent update for $\{\theta, \Phi\}$) to reach at least a local maximum of ELBO. To better approximate the optimal discriminator, we perform multiple calls of Algorithm 2 in every iteration of the main loop in Algorithm 1 and also apply a larger learning rate α_Ψ. We have observed in our own experiments that these tricks improve the predictive performance of IPVI.

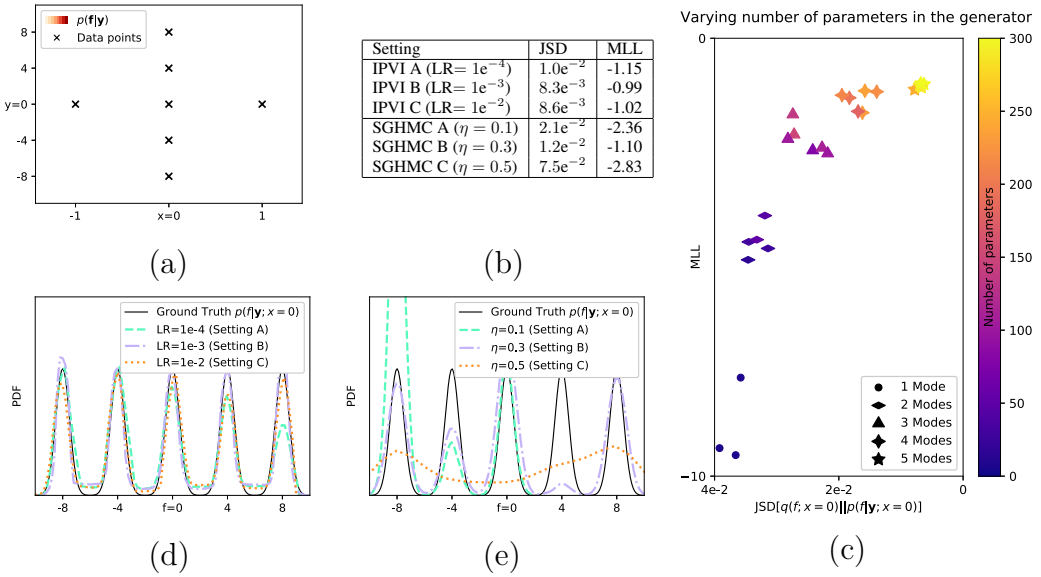
4.4 Experiments and Discussion

In this section, we empirically evaluate and compare the performance of our IPVI framework against that of the state-of-the-art SGHMC [Havasi *et al.*, 2018] and *doubly stochastic VI* [4] (DSVI) [Salimbeni and Deisenroth, 2017] for DGPs based on their publicly available implementations using synthetic and real-world datasets in supervised (e.g., regression and classification) and unsupervised learning tasks.

4.4.1 Synthetic Experiment: Learning a Multi-Modal Posterior Belief

To demonstrate the capability of IPVI in learning a complex multi-modal posterior belief, we generate a synthetic "diamond" dataset and adopt a multi-modal mixture of Gaussian prior belief $p(\mathbf{f})$ (see Appendix C.4.1 for its description) to yield a multi-modal posterior belief $p(\mathbf{f}|\mathbf{y})$ for a single-layer GP. Fig. 4.2a illustrates this dataset and ground-truth posterior belief. Specifically, we focus on the multi-modal posterior belief $p(f|\mathbf{y}; x = 0)$ at input $x = 0$ whose ground truth is shown in Fig. 4.2d. Fig. 4.2c shows that as the number of parameters in the generator increases, the expressive power of IPVI increases such that its variational posterior $q(f; x = 0)$ can capture

[4] It is reported in [Salimbeni and Deisenroth, 2017] that DSVI has outperformed the approximate expectation propagation method of [Bui *et al.*, 2016] for DGPs. Hence, we do not empirically compare with the latter [Bui *et al.*, 2016] here.



Setting	JSD	MLL
IPVI A (LR= $1e^{-4}$)	$1.0e^{-2}$	-1.15
IPVI B (LR= $1e^{-3}$)	$8.3e^{-3}$	-0.99
IPVI C (LR= $1e^{-2}$)	$8.6e^{-3}$	-1.02
SGHMC A ($\eta = 0.1$)	$2.1e^{-2}$	-2.36
SGHMC B ($\eta = 0.3$)	$1.2e^{-2}$	-1.10
SGHMC C ($\eta = 0.5$)	$7.5e^{-2}$	-2.83

Figure 4.2: (a) The *probability density function* (PDF) plot of the ground-truth posterior belief $p(\mathbf{f}|\mathbf{y})$. (b) Performances of IPVI and SGHMC in terms of estimated *Jenson-Shannon divergence* (JSD) and *mean log-likelihood* (MLL) metrics under the respective settings of varying learning rates α_Ψ and step sizes η. (c) Graph of MLL vs. JSD achieved by IPVI with varying number of parameters in the generator: Different shapes indicate varying number of modes learned by the generator. (d-e) PDF plots of variational posterior $q(f; x = 0)$ learned using (d) IPVI with generators of varying learning rates α_Ψ and (e) SGHMC with varying step sizes η.

more modes in the true posterior, thus resulting in a closer estimated *Jensen-Shannon divergence* (JSD) between them and a higher *mean log-likelihood* (MLL).

Next, we compare the robustness of IPVI and SGHMC in learning the true multi-modal posterior belief $p(f|\mathbf{y}; x = 0)$ under different hyperparameter settings[5]: The generators in IPVI use the same architecture with about 300 parameters but different learning rates α_Ψ, while the SGHMC samplers use different step sizes η. The results in Figs. 4.2b and 4.2e have verified a remark made in [Zhang *et al.*, 2019] that SGHMC

[5]We adopt scale-adapted SGHMC which is a robust variant used in Bayesian neural networks and DGP inference [Havasi *et al.*, 2018]. A recent work of [Zhang *et al.*, 2019] has proposed the cyclical stochastic gradient MCMC method to improve the accuracy of sampling highly complex distributions. However, it is not obvious to us how this method can be incorporated into DGP models, which is beyond the scope of this work.

is sensitive to the step size which cannot be set automatically [Springenberg *et al.*, 2016] and requires some prior knowledge to do so: Sampling with a small step size is prone to getting trapped in local modes while a slight increase of the step size may lead to an over-flattened posterior estimate. Additional results for different hyperparameter settings of SGHMC can be found in Appendix C.4.1 In contrast, the results in Figs. 4.2b and 4.2d reveal that, given enough parameters, IPVI performs robustly under a wide range of learning rates.

4.4.2 Supervised Learning: Regression and Classification

For our experiments in the regression tasks, the depth L of the DGP models are varied from 1 to 5 with 128 inducing inputs per layer. The dimension of each hidden DGP layer is set to be (a) the same as the input dimension for the UCI benchmark regression and Airline datasets, (b) 16 for the YearMSD dataset, and (c) 98 for the classification tasks. The experimental settings for supervised learning tasks are found in Appendix C.4.2:

4.4.2.1 Regression

UCI Benchmark Regression. Our experiments are first conducted on 7 UCI benchmark regression datasets. We have performed a random 0.9/0.1 train/test split.

Large-Scale Regression. We then evaluate the performance of IPVI on two real-world large-scale regression datasets: (a) YearMSD dataset with a large input dimension $D = 90$ and data size $N \approx 500000$, and (b) Airline dataset with input dimension $D = 8$ and a large data size $N \approx 2$ million. For YearMSD dataset, we use the first 463715 examples as training data and the last 51630 examples as test data[6]. For Airline dataset, we set the last 100000 examples as test data.

[6]This avoids the 'producer' effect by ensuring that no song from an artist appears in both training & test data.

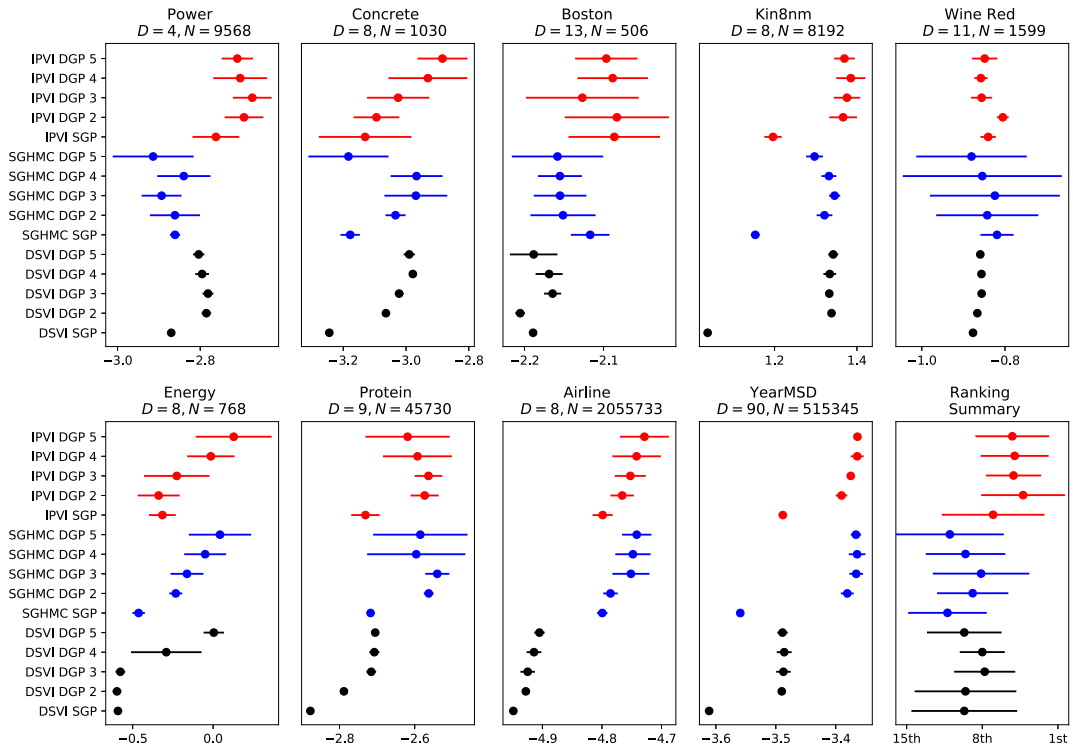


Figure 4.3: Mean test log-likelihood and standard deviation achieved by our IPVI framework (red), SGHMC (blue), and DSVI (black) for DGPs for UCI benchmark and large-scale regression datasets. Higher test log-likelihood (i.e., to the right) is better.

In the above regression tasks, the performance metric is the *mean test log-likelihood* (MLL). Fig. 4.3 shows results of the mean test log-likelihood and standard deviation over 10 runs. It can be observed that IPVI generally outperforms SGHMC and DSVI and the ranking summary shows that our IPVI framework for a 2-layer DGP model (IPVI DGP 2) performs the best on average across all regression tasks. For large-scale regression tasks, the performance of IPVI consistently increases with a greater depth. Even for a small dataset, the performance of IPVI improves up to a certain depth.

Remark 9. As can be observed from Figure 4.3, DSVI DGP appears to perform the worst among these three models. Interestingly, standard deviation for DSVI DGP is almost invisible, this observation collaborates with our claim in Section 2.1 that VI

is prone to underestimate the uncertainty of the prediction.

4.4.2.2 Classification

We evaluate the performance of IPVI in three classification tasks using the real-world MNIST, fashion-MNIST, and CIFAR-10 datasets. Both MNIST and fashion-MNIST datasets are grey-scale images of 28×28 pixels. The CIFAR-10 dataset consists of colored images of 32×32 pixels. We utilize a 4-layer DGP model with 100 inducing inputs per layer and a robust-max multiclass likelihood [Hernández-Lobato *et al.*, 2011].

Real World Classification. Table 4.1 reports the mean test accuracy over 10 runs, which shows that our IPVI framework for a 4-layer DGP model performs the best in all three datasets.

Table 4.1: Mean test accuracy (%) achieved by IPVI, SGHMC, and DSVI for 3 classification datasets.

Dataset	MNIST		Fashion-MNIST		CIFAR-10	
	SGP	DGP 4	SGP	DGP 4	SGP	DGP 4
DSVI	**97.32**	97.41	86.98	87.99	47.15	51.79
SGHMC	96.41	97.55	85.84	87.08	47.32	52.81
IPVI	97.02	**97.80**	**87.29**	**88.90**	**48.07**	**53.27**

Convolutional Skip-layer Connection (CSC): For the image datasets, the data distribution has local correlation between pixels. It would be better if the skip-layer connection can incorporate such information as a base for invariant mapping. We change the skip-layer connection from a fully connected manner to a convolutional manner, which is described in detail in Appendix C.4.2. Results show that the CSC boosts the DGP performance in real world image datasets and performs best when integrated with IPVI.

Table 4.2: Mean test accuracy (%) achieved by IPVI, SGHMC, and DSVI for 3 classification datasets with convolutions.

Dataset	MNIST		Fashion-MNIST		CIFAR-10	
	SGP	DGP 4	SGP	DGP 4	SGP	DGP 4
DSVI	**97.32**	99.16	86.98	91.57	47.15	75.05
SGHMC	96.41	98.15	85.84	88.14	47.32	70.78
IPVI	97.02	**99.32**	**87.29**	**91.78**	**48.07**	**76.11**

4.4.3 Unsupervised Learning: FreyFace Reconstruction

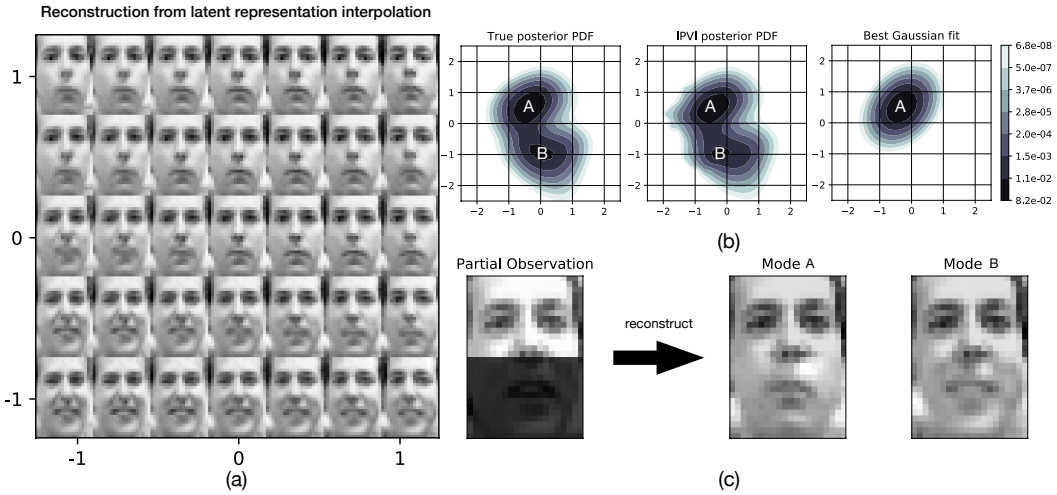


Figure 4.4: Unsupervised learning with FreyFace dataset. (a) Latent representation interpolation and the corresponding reconstruction. (b) True posterior $p(\mathbf{x}^\star|\mathbf{y}_O^\star)$ given the partial observation $\mathbf{y}_O^\star$ (left), variational posterior $q(\mathbf{x}^\star)$ learned by IPVI (middle), and Gaussian approximation (right). The PDF for $p(\mathbf{x}^\star|\mathbf{y}_O^\star)$ is calculated using Bayes rule where the marginal likelihood is computed using Monte Carlo integration. (c) The partial observation (with the ground truth reflected in the dark region) and two reconstructed samples from $q(\mathbf{x}^\star)$.

A DGP can naturally be generalized to perform unsupervised learning. The representation of a dataset in a low-dimensional manifold can be learned in an unsupervised manner by the *GP latent variable model* (GPLVM) [Lawrence, 2004] where only the observations $\mathbf{Y} \triangleq \{\mathbf{y}_n\}_{n=1}^N$ are given and the hidden representation $\mathbf{X}$ is unobserved and treated as latent variables. The objective is to infer the posterior $p(\mathbf{X}|\mathbf{Y})$. The

GPLVM is a single-layer GP that casts $\mathbf{X}$ as an unknown distribution and can naturally be extended to a DGP. So, we construct a 2-layer DGP ($\mathbf{X} \to \mathbf{F}_1 \to \mathbf{F}_2 \to \mathbf{Y}$) and use the generator samples to represent $p(\mathbf{X}|\mathbf{Y})$.

We consider the FreyFace dataset [Roweis *et al.*, 2002] taken from a video sequence that consists of 1965 images with a size of 28×20. We select the first 1000 images to train our DGP. To ease visualization, the dimension of latent variables $\mathbf{X}$ is chosen to be 2. Additional details for our experiments are found in Appendix C.4.3. Fig. 4.4a shows the reconstruction of faces across the latent space. Interestingly, the first dimension of the latent variables $\mathbf{X}$ determines the expression from happy to calm while the the second dimension controls the view angle of the face.

We then explore the capability of IPVI in reconstructing partially observed test data. Fig. 4.4b illustrates that given only the upper half of the face, the real face may exhibit a multi-modal property, as reflected in the latent space; intuitively, one cannot always tell whether a person is happy or sad by looking at the upper half of the face. Our variational posterior accurately captures the multi-modal posterior belief whereas the Gaussian approximation can only recover one mode (mode A) under this test scenario. So, IPVI can correctly recover two types of expressions: calm (mode A) and happy (mode B). We did not empirically compare with SGHMC here because it is not obvious to us whether their sampler setting can be carried over to this unsupervised learning task.

4.4.4 Time Efficiency

Table 4.3 and Fig. 4.5 show the better time efficiency of IPVI over the state-of-the-art SGHMC for a 4-layer DGP model that is trained using the Airline dataset. The learning rates are 0.005 and 0.02 for IPVI and SGHMC (default setting adopted from [Havasi *et al.*, 2018]), respectively. Due to parallel sampling (Section 4.3), our

IPVI framework enables posterior samples to be generated 500 times faster. Although IPVI has more parameters than SGHMC, it runs 9 times faster during training due to efficiency in sample generation.

Table 4.3: Time incurred by a 4-layer DGP model for Airline dataset.

	IPVI	SGHMC
Average training time (per iter.)	0.35 sec.	3.18 sec.
$\boldsymbol{\mathcal{U}}$ generation (100 samples)	0.28 sec.	143.7 sec.

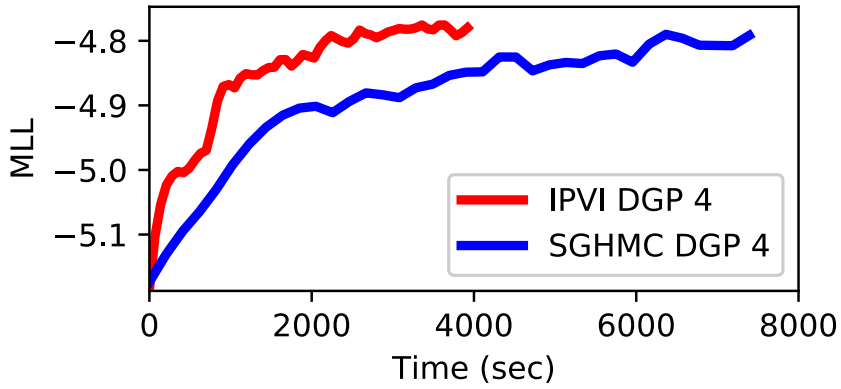


Figure 4.5: Graph of MLL vs. total incurred time to train a 4-layer DGP model for the Airline dataset.

4.5 Summary

This chapter describes a novel IPVI framework for DGPs that can ideally recover an unbiased posterior belief of the inducing variables and still preserve the time efficiency of VI. To achieve this, we cast the DGP inference problem as a two-player game and search for a Nash equilibrium (i.e., an unbiased posterior belief) of this game using best-response dynamics. We propose a novel parameter-tying architecture of the generator and discriminator in our IPVI framework for DGPs to alleviate overfitting and speed up training and prediction. Empirical evaluation shows that

IPVI outperforms the state-of-the-art approximation methods for DGPs in regression and classification tasks and accurately learns complex multi-modal posterior beliefs in our synthetic experiment and an unsupervised learning task.

Chapter 5

Convolutional Normalizing Flows for Deep Gaussian Processes

This chapter is based on the paper submitted to the 2020 International Conference on Machine Learning (ICML-20): "Variational Inference for Deep Gaussian Processes with Convolutional Normalizing Flows".

5.1 Introduction

In Chapter 4, we discussed thoroughly an novel inference framework, IPVI, for DGPs which can ideally recover the unbiased posterior belief as well as preserve the time efficiency of VI. To achieve this, we cast the DGP inference problem as a two-player game and search for a Nash equilibrium using BRD. However, the work of [Goodfellow, 2016; Salimans *et al.*, 2016] pointed out the critical issue of *convergence*. In fact, Remark 8 also mentioned that there is no guarantee that BRD converges to a Nash equilibrium, hence giving no assurance of recovering the true posterior distribution.

To resolve the intrinsic issues related with IPVI, this chapter proposes a novel framework based on the idea of *normalizing flows* [Tabak and Vanden-Eijnden, 2010;

Tabak and Turner, 2013] to model the complex posterior distribution directly. More interestingly, iin contrast to IPVI in which the posterior distribution is represented through samples, normalizing flows allow for exact and efficient evaluation of the density of the posterior distribution. In this regard, this chapter firstly reviews the technical details of NF in Section 5.2 and Section 5.3. Next, Section 5.4 and Section 5.5 explain the detailed architecture design of NF targeted at modeling DGP posterior distribution efficiently and effectively. Lastly, empirical evaluations demonstrate DGP with NF outperforms existing approximation methods for DGP models.

5.2 Normalizing Flows

A normalizing flow is a transformation of a simple distribution $\pi(\mathbf{z})$ into a complex distribution $p(\mathbf{x})$ by applying a sequence of invertible mappings. Figure 5.1 illustrates a transformation from a standard Gaussian distribution to a mixture of Gaussian distributions through a function f (Note that since f is invertible, g represents the inverse function of f, denoted as $g = f^{-1}$).

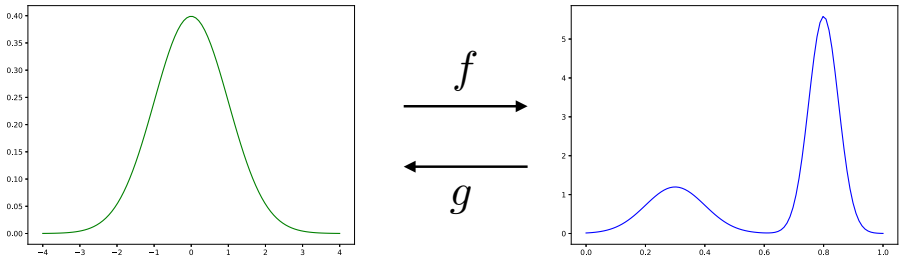


Figure 5.1: An illustration of normalizing flows. Left: the density of the simple distribution $\pi(\mathbf{z})$. Right: the density of the complex distribution $p(\mathbf{x})$.

Following the *change of variable* rule, the normalizing flow framework can be defined as:

Definition 1. Let a random variable $\mathbf{z} \in \mathbb{R}^D$ follow a distribution: $\mathbf{z} \sim \pi(\mathbf{z})$. Given a

bijective function $f : \mathbb{R}^D \to \mathbb{R}^D$ and a new random variable $\mathbf{x} = f(\mathbf{z})$, the probability density function of $p(\mathbf{x})$ can be written as:

$$\begin{aligned} p(\mathbf{x}) &= \pi(\mathbf{z})|\det(\frac{\mathrm{d}\mathbf{z}}{\mathrm{d}\mathbf{x}})| \\ p(\mathbf{x}) &= \pi(\mathbf{z})|\det(\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{z}})|^{-1} \\ p(\mathbf{x}) &= \pi(\mathbf{z})|\det(f^{'}(\mathbf{z}))|^{-1} \end{aligned} \tag{5.1}$$

where $|\det(f^{'}(\mathbf{z}))|$ denotes the absolute value of the determinant of the Jacobian of f evaluated at $\mathbf{z}$.

Remark 10. Intuitively, if the transformation function f can be arbitrarily complex, we can transform the simple distribution $\pi(\mathbf{z})$ into any complex distribution $p(\mathbf{x})$. We refer the readers to [Villani, 2003; Bogachev *et al.*, 2005; Medvedev, 2008] for a detailed discussion of the proof.

5.3 Density Estimation and Density Matching

5.3.1 Density Estimation

Taking a closer inspection of Figure 5.1, we can transform a simple distribution into a complex distribution and vice versa. Naturally, it makes normalizing flows a good fit to perform density estimation as shown in Figure 5.2.

Without loss of generality, we assume data $\mathbf{x}$ are sampled from some unknown distribution $p(\mathbf{x})$ (e.g. images of human faces) and our goal is to transform them into a simple distribution $\pi(\mathbf{z})$ (e.g. standard Gaussian distribution) through $g : \mathbf{x} \to \mathbf{z}$.

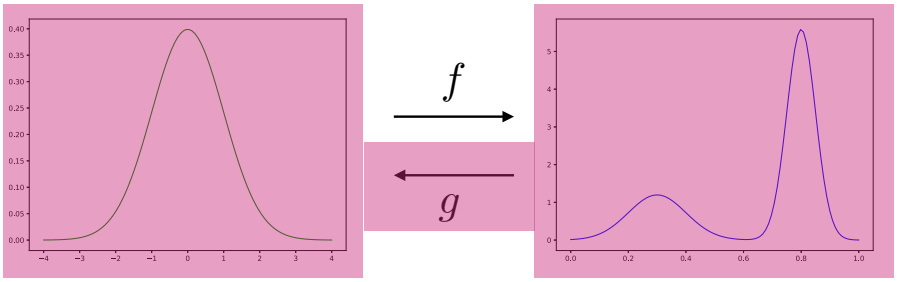


Figure 5.2: An illustration of density estimation. The objective is to transform a complex distribution $p(\mathbf{x})$ into a simple distribution $\pi(\mathbf{z})$

According to Definition 1, we have:

$$\begin{aligned} \log \pi(\mathbf{z}) &= \log p(\mathbf{x}) - \log |\frac{\mathrm{d}g}{\mathrm{d}\mathbf{x}}| \\ \log p(\mathbf{x}) &= \log \pi(\mathbf{z}) + \log |\frac{\mathrm{d}g}{\mathrm{d}\mathbf{x}}| \\ \log p(\mathbf{x}) &= \log \pi(g(\mathbf{x})) + \log |\frac{\mathrm{d}g}{\mathrm{d}\mathbf{x}}| \end{aligned} \tag{5.2}$$

Hence, our objective is to maximize the log data likelihood $\log p(\mathbf{x})$. Note that density estimation with normalizing flows has been a building block for modern generative modeling [Kingma and Dhariwal, 2018; Dinh and Bengio, 2016].

5.3.2 Relations with Expectation Propagation

Interestingly, density estimation can be considered as an EP (Section 2.1.2) problem. Suppose we have a random variable $\mathbf{x}$, and we only have access to the samples drawn from the unknown target distribution $P(\mathbf{x})$ (e.g. the unknown distribution of images) but lack direct access to $P(\mathbf{x})$. Our objective is to construct a distribution $p(\mathbf{x}) = P(\mathbf{x})$. Therefore, we would like to minimize the KL distance between the target

distribution $P(\mathbf{x})$ and the constructed distribution $p(\mathbf{x})$:

$$\begin{aligned}\min \mathrm{KL}(P(\mathbf{x})||p(\mathbf{x})) &= \min_p -\mathbb{H}_P[\mathbf{x}] - \mathbb{E}_{P(\mathbf{x})}[\log p(\mathbf{x})] \\ &= \min_p \underbrace{-\mathbb{H}_P[\mathbf{x}]}_{constant} - \mathbb{E}_{P(\mathbf{x})}[\log p(\mathbf{x})] \\ &= \min_p -\mathbb{E}_{P(\mathbf{x})}[\log p(\mathbf{x})] + \text{const.}\end{aligned} \tag{5.3}$$

According to (5.2), the equation above can be written as:

$$\min_g -\mathbb{E}_{P(\mathbf{x})}\Big[\log \pi(g(\mathbf{x})) + \log |\frac{\mathrm{d}g}{\mathrm{d}\mathbf{x}}|\Big] \tag{5.4}$$

Remark 11. Note that minimizing the KL distance in (5.3) is equivalent to maximizing the log likelihood of the data.

5.3.3 Density Matching and Variational Inference

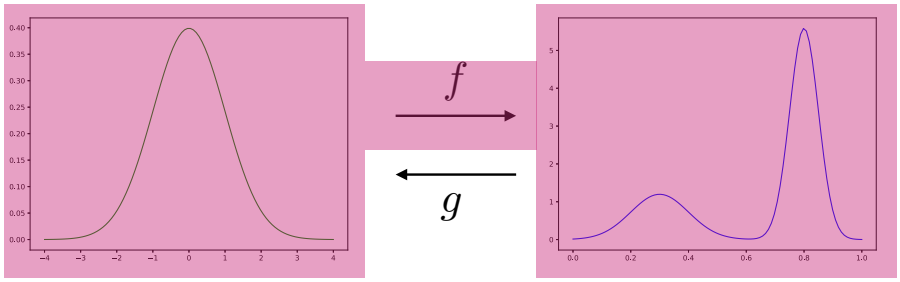


Figure 5.3: An illustration of density matching. The objective is to represent a complex distribution $p(\mathbf{x})$ with a simple distribution $\pi(\mathbf{z})$

Figure 5.3 shows the reverse scenario where we have access to the target distribution $P(\mathbf{x})$ with its analytical form but have no available sampling procedure (e.g. $P(\mathbf{x})$ is an unnormalized distribution). Our objective is to build a normalizing flow $f : \mathbf{z} \to \mathbf{x}$ which transforms a simple distribution $\pi(\mathbf{z})$ to a complex distribution $p(\mathbf{x})$

by minimizing the KL distance between $p(\mathbf{x})$ and $P(\mathbf{x})$:

$$\begin{aligned}\min \mathrm{KL}(p(\mathbf{x})||P(\mathbf{x})) &= \min \mathbb{E}_{p(\mathbf{x})}[\log p(\mathbf{x}) - \log P(\mathbf{x})] \\ &= \min \mathbb{E}_{\pi(\mathbf{z})}[\log \pi(\mathbf{z}) - \log |\frac{\mathrm{d}g}{\mathrm{d}\mathbf{z}}| - \log P(g(\mathbf{z}))]\end{aligned} \tag{5.5}$$

Remark 12. Rezende and Mohamed (2015) were the first to propose the utilization of normalizing flows in the context of VI. Interestingly, as we have discussed in Section 1.1.2, the posterior distribution is proportional to the joint distribution, which suggests that the posterior distribution is an unnormalized version of the joint distribution. Therefore, given the joint distribution, the true posterior distribution can be recovered using normalizing flow.

5.4 Variational Inference for DGPs with Normalizing Flows

In this section, we propose a novel variational inference framework for DGPs utilizing the notion of normalizing flows. Recall from Section 4.2 that in a DGP with a depth of L, each DGP layer is associated with a set $\mathbf{F}_{\ell-1}$ of inputs and a set $\mathbf{F}_\ell$ of outputs for $\ell = 1, \ldots, L$ and $\mathbf{F}_0 \triangleq \mathbf{X}$. Let $\boldsymbol{\mathcal{F}} \triangleq \{\mathbf{F}_\ell\}_{\ell=1}^L$, and let the inducing inputs and corresponding inducing output variables for DGP layers $\ell = 1, \ldots, L$ be denoted by the respective sets $\boldsymbol{\mathcal{Z}} \triangleq \{\mathbf{Z}_\ell\}_{\ell=1}^L$ and $\boldsymbol{\mathcal{U}} \triangleq \{\mathbf{U}_\ell\}_{\ell=1}^L$. The joint probability distribution of the DGP model is:

$$p(\mathbf{y}, \boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{U}}) = \underbrace{p(\mathbf{y}|\mathbf{F}_L)}_{\text{data likelihood}} \underbrace{\left[\prod_{\ell=1}^{L} p(\mathbf{F}_\ell|\mathbf{U}_\ell)\right] p(\boldsymbol{\mathcal{U}})}_{\text{DGP prior}}.$$

Similarly, the variational posterior is assumed to be:

$$q(\boldsymbol{\mathcal{F}},\boldsymbol{\mathcal{U}}) = \left[\prod_{\ell=1}^{L} p(\mathbf{F}_\ell|\mathbf{U}_\ell)\right] q(\boldsymbol{\mathcal{U}}). \tag{5.6}$$

Unlike the IPVI framework in Section 4.3, the variational posterior $q(\boldsymbol{\mathcal{U}})$ is constructed through a normalizing flow $\mathcal{G} : \boldsymbol{\mathcal{V}} \to \boldsymbol{\mathcal{U}}$ where $\boldsymbol{\mathcal{V}} \sim \pi(\cdot)$ is a new random variable with distribution $\pi(\cdot)$.

In accordance with Section 5.3.3, our objective is to minimize the KL distance $\mathrm{KL}(q(\boldsymbol{\mathcal{U}})||p(\boldsymbol{\mathcal{U}}|\mathbf{y}))$. Following the Bayes' theorem, the true posterior distribution $p(\boldsymbol{\mathcal{U}}|\mathbf{y})$ can be written as:

$$p(\boldsymbol{\mathcal{U}}|\mathbf{y}) = \frac{p\left(\mathbf{y}|\mathbf{F},\boldsymbol{\mathcal{U}}\right) p(\boldsymbol{\mathcal{U}})}{p\left(\mathbf{y}|\mathbf{F}\right)} \tag{5.7}$$

where $\mathbf{F}$ is a Monte Carlo sample from the predictive distribution of the layer outputs: $\mathbf{F} \sim \prod_{\ell=1}^{L} p(\mathbf{F}_\ell|\mathbf{U}_\ell, \mathbf{F}_{\ell-1})$[1]. Therefore, the KL distance can be written as:

$$\begin{aligned}
\mathrm{KL}(q(\boldsymbol{\mathcal{U}})||p(\boldsymbol{\mathcal{U}}|\mathbf{y})) &= \mathrm{KL}(q(\boldsymbol{\mathcal{U}})||p(\mathbf{y}|\mathbf{F},\boldsymbol{\mathcal{U}})p(\boldsymbol{\mathcal{U}})/p(\mathbf{y}|\mathbf{F})) \\
&= \int q(\boldsymbol{\mathcal{U}}) \left[\log \frac{q(\boldsymbol{\mathcal{U}})}{p(\mathbf{y}|\mathbf{F},\boldsymbol{\mathcal{U}})p(\boldsymbol{\mathcal{U}})}\right] \mathrm{d}\boldsymbol{\mathcal{U}} + \log p(\mathbf{y}|\mathbf{F}) \\
&= \mathbb{E}_{q(\boldsymbol{\mathcal{U}})}\left[\log q(\boldsymbol{\mathcal{U}}) - \log p(\mathbf{y}|\mathbf{F},\boldsymbol{\mathcal{U}}) - p(\boldsymbol{\mathcal{U}})\right] + \log p(\mathbf{y}|\mathbf{F}) \\
&= \mathbb{E}_{\pi(\boldsymbol{\mathcal{V}})}[\log \pi(\boldsymbol{\mathcal{V}}) - \log |\frac{\mathrm{d}\mathcal{G}}{\mathrm{d}\boldsymbol{\mathcal{V}}}| - \log p(\mathbf{y}|\mathbf{F},\mathcal{G}(\boldsymbol{\mathcal{V}})) - \log p(\mathcal{G}(\boldsymbol{\mathcal{V}}))] + \text{const}
\end{aligned} \tag{5.8}$$

[1]Note that to compute $\mathbf{F}_L$, we still adopt the same sampling method proposed by [Salimbeni and Deisenroth, 2017].

5.5 Normalizing Flow with Convolutions

In this section, we discuss how the architecture of the normalizing flow is designed for DGP. Recall from Section 5.4 that $\boldsymbol{\mathcal{U}} = \{\mathbf{U}_\ell \in \mathbb{R}^{M \times d_\ell}\}_{\ell=1}^{L}$ is a collection of inducing variables for DGP layers $\ell = 1, \ldots, L$. M represents the number of inducing output variables, and d_ℓ denotes the dimension of inducing output variables for layer ℓ.

A naive design is to consider a layer-wise normalizing flow which is illustrated in Figure 5.4. However, such a naive design suffers from the following critical issues:

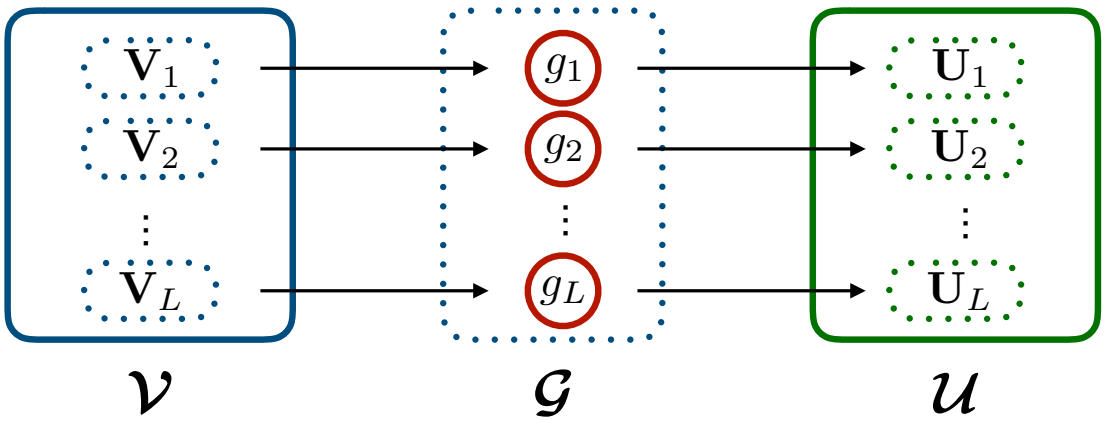


Figure 5.4: A naive design of normalizing flow for DGP. The normalizing flow $\boldsymbol{\mathcal{G}}$ is separated into L individual flows.

- Figure 5.4 shows that to recover the posterior samples of M different inducing variables $\mathbf{u}_{\ell 1}, \ldots, \mathbf{u}_{\ell M}$ ($\mathbf{U}_\ell = \{\mathbf{u}_{\ell 1}, \ldots, \mathbf{u}_{\ell M}\}$), it is natural to design the normalizing flow $g_\ell : \mathbf{R}^{Md_\ell} \to \mathbf{R}^{Md_\ell}$. Therefore, a large number of parameters is needed, which will increase the risk of overfitting.

- Another critical issue is the computational complexity. In general, computing the log Jacobian determinant incurs $\mathcal{O}(M^3 \cdot d_\ell^3)$, hence resulting in the difficulty in optimization.

- Such a design fails to capture the dependency of the inducing output variables $\mathbf{U}_\ell$ among different layers. As pointed out by [Yu *et al.*, 2019a], the posterior distribution of $p(\boldsymbol{\mathcal{U}}|\mathbf{y})$ is highly correlated among layers.

- Such a design fails to adequately capture the dependency of the inducing output variables $\mathbf{U}_\ell$ on its corresponding inducing inputs $\mathbf{Z}_\ell$, hence restricting its capability to model the output posterior $\boldsymbol{\mathcal{U}}$ accurately.

To resolve the above issues, we propose a novel normalizing flow architecture with convolution for DGP models, as shown in Figure 5.5. Instead of treating $\boldsymbol{\mathcal{V}} = \{\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_L\}$ separately, we decide to stack $\{\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_L\}$ to be a three-dimensional tensor denoted as:

$$\boldsymbol{\mathcal{V}} \in \mathbb{R}^{M \times 1 \times \sum_{\ell=1}^{L} d_\ell} \tag{5.9}$$

and design our normalizing flow $\boldsymbol{\mathcal{G}} : \mathbb{R}^{M \times 1 \times \sum_{\ell=1}^{L} d_\ell} \to \mathbb{R}^{M \times 1 \times \sum_{\ell=1}^{L} d_\ell}$ accordingly as shown in Figure 5.5.

To this end, we propose to convolve $\boldsymbol{\mathcal{V}}$ with a kernel tensor $\mathbf{W}$ where $\mathbf{W} \in \mathbb{R}^{1 \times 1 \times \sum_{\ell=1}^{L} d_\ell \times \sum_{\ell=1}^{L} d_\ell}$ as shown in Figure 5.5. In this way, the normalizing flow $\mathcal{G}$ is fully characterized by the kernel tensor $\mathbf{W}$ with a significantly smaller number of parameters. Hence the log Jacobian determinant can be easily written as:

$$\log \left| \det \left(\frac{\mathrm{d}\, \boldsymbol{\mathcal{G}}}{\mathrm{d}\, \boldsymbol{\mathcal{V}}} \right) \right| = M \times 1 \times \log |\det(\mathbf{W})| \tag{5.10}$$

Remark 13. Note that compared with the naive design in Figure 5.4, computing the log Jacobian determinant only incurs $\mathcal{O}(\left(\sum_{\ell=1}^{L} d_\ell \right)^3)$, which reduces the time complexity by $\mathcal{O}(M^3)$ times. Moreover, compared with the naive design, it is also easier to compute the determinant of $\mathbf{W}$. Another advantage is that the kernel tensor $\mathbf{W}$ naturally captures the dependency of inducing variables $\{\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_L\}$ between

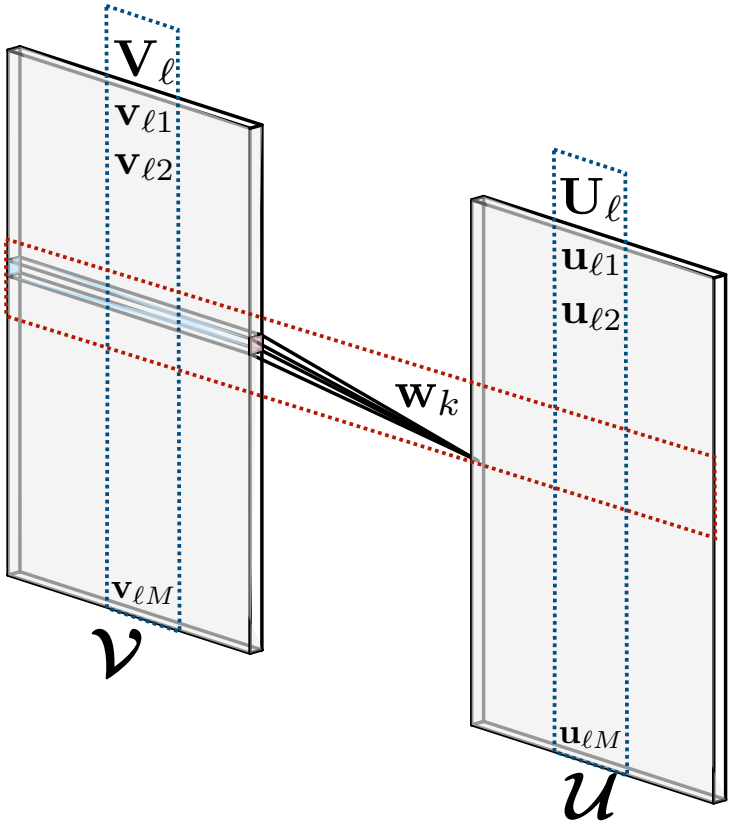


Figure 5.5: The normalizing flow with convolution for DGP. The kernel tensor $\mathbf{W}$ can be decomposed into a set of tensors $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K\}$ where $\mathbf{w}_1 \in \mathbb{R}^{1\times 1\times \sum_{\ell=1}^{L} d_\ell}$ $K = \sum_{\ell=1}^{L} d_\ell$. The red box indicates the convolution with $\mathbf{w}_k$.

layers.

Furthermore, to capture the dependency of the inducing output variables $\mathbf{U}_\ell$ on its corresponding inducing inputs $\mathbf{Z}_\ell$, we manually construct the base distribution $\pi(\boldsymbol{\mathcal{V}})$ to depend on the inducing inputs $\boldsymbol{\mathcal{Z}}$. For each layer ℓ, the base distribution can be written:

$$\mathbf{V}_\ell \sim \mathcal{N}(\boldsymbol{\mu}_\ell, \boldsymbol{\sigma}_\ell), \text{ where } [\boldsymbol{\mu}_\ell, \boldsymbol{\sigma}_\ell] = \varphi_\ell(\mathbf{Z}_\ell) \tag{5.11}$$

where φ_ℓ is a neural network.

We observe from our experiments that our normalizing flow for DGP with convo-

lutions improves the performance considerably, which will be shown in Section 5.6.

5.6 Experiments and Discussion

This section empirically evaluates the performance of the normalizing flow with convolution (NF) framework. Moreover, we compare the performance with IPVI, SGHMC, and DSVI for DGPs on real-world regression and classification datasets.

5.6.1 Synthetic Experiments: 2D Multi-modal Density Matching

To demonstrate the representation power of our proposed normalizing flow framework, we manually parametrized a set of unnormalized 2D multi-modal density functions[2]:

$$p(\mathbf{z}) \propto \exp(-U(\mathbf{z})) \tag{5.12}$$

Figure 5.6 shows that our proposed normalizing flow framework can accurately capture the true multi-model probability density. More interestingly, unlike the IVPI method in [Yu *et al.*, 2019a] or SGHMC method in [Havasi *et al.*, 2018] which can only represent the distribution through samples; our proposed normalizing flow method can also provide analytical evaluation of the intractable probability density function (PDF) which is one of the most important properties in evaluating a distribution.

[2]please refer to the Appendix D.1 for details of the energy function $U(\cdot)$

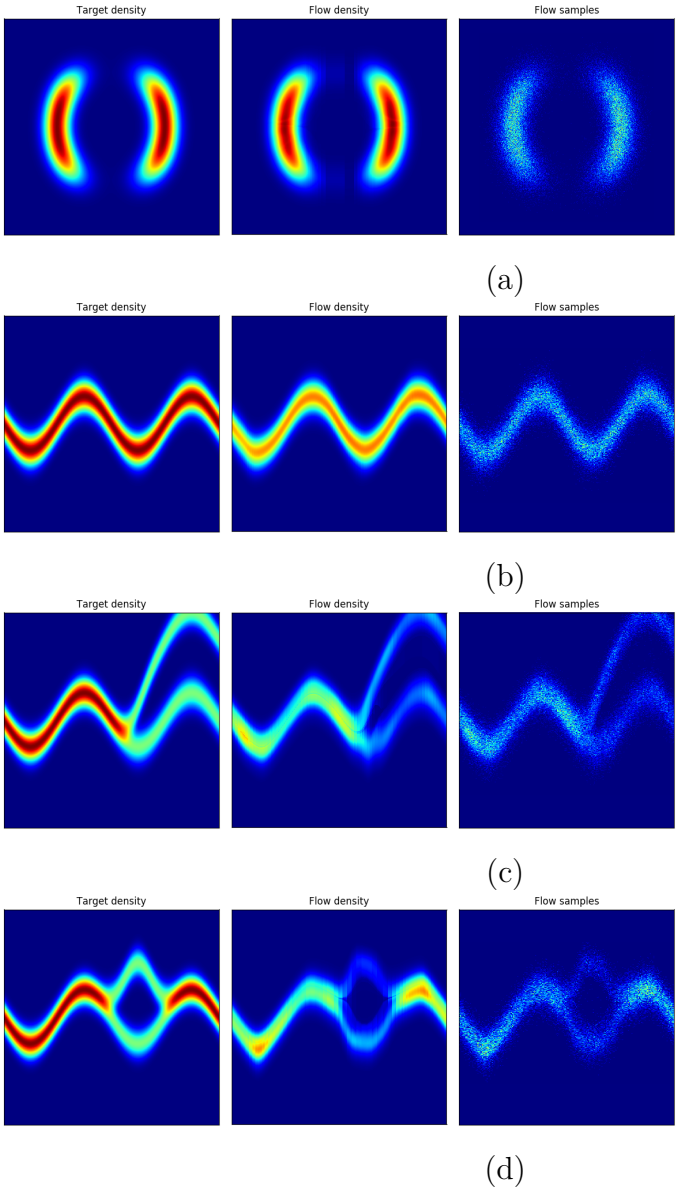


Figure 5.6: 4 unnormalized density functions. Left column shows the unnormalized distribution for these four cases; middle column illustrates the performance of our normalizing flow framework in approximating these four distributions; right column demonstrates the samples drawn from our normalizing flow framework.

5.6.2 Supervised Learning: Regression and Classification

For our experiments in the regression tasks, the depth L of the DGP models are varied from 2 to 5 with 128 inducing inputs per layer. The dimension of each hidden DGP layer is set to be (a) the same as the input dimension for the UCI benchmark regression and Airline datasets, (b) 16 for the YearMSD dataset, and (c) 98 for the classification tasks. Please refer to Appendix D.2 for more details about the experimental settings.

5.6.2.1 Regression

UCI Benchmark Regression. Our experiments are first conducted on 7 UCI benchmark regression datasets. We have performed a random 0.9/0.1 train/test split.

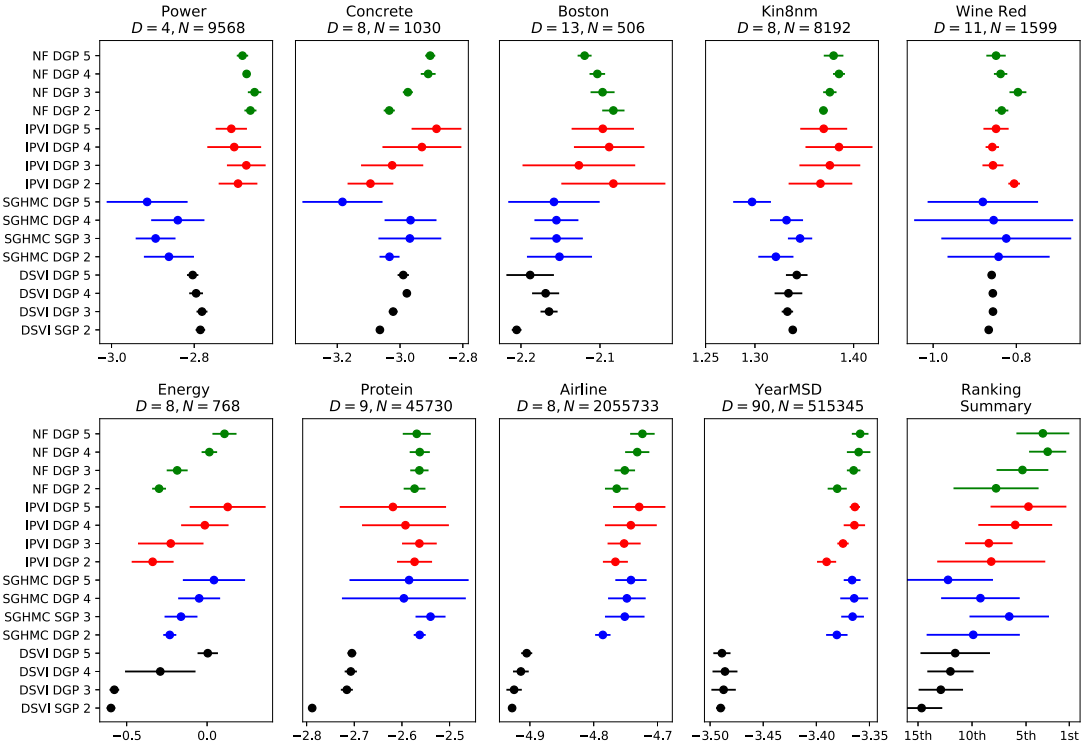


Figure 5.7: Mean test log-likelihood and standard deviation achieved by our NF framework (green), IPVI (red), SGHMC (blue), and DSVI (black) for DGPs for UCI benchmark and large-scale regression datasets. Higher test log-likelihood (i.e., to the right) is better.

Large-Scale Regression. We then evaluate the performance of NF on two real-world large-scale regression datasets: (a) YearMSD dataset with a large input dimension $D = 90$ and data size $N \approx 500000$, and (b) Airline dataset with input dimension $D = 8$ and a large data size $N \approx 2$ million. For YearMSD dataset, we use the first 463715 examples as training data and the last 51630 examples as test data. For Airline dataset, we set the last 100000 examples as test data.

In the above regression tasks, the performance metric is the *mean test log-likelihood* (MLL). Figure 5.7 shows the results of the mean test log-likelihood and standard deviation over 10 runs. It can be observed that NF generally outperforms other frameworks and the ranking summary shows that our NF framework for a 4-layer DGP model (NF DGP 4) performs the best on average across all regression tasks. For large-scale regression tasks, the performance of NF consistently increases with a greater depth.

Similar to DSVI, it can be observed from Figure 5.7, the results of NF DGP is very consistent across different runs (as can be seen from the smaller standard deviations) compared with SGHMC and IPVI which represent the posterior distribution with samples.

5.6.2.2 Classification

We evaluate the performance of IPVI in three classification tasks using the real-world MNIST, fashion-MNIST, and CIFAR-10 datasets. Both MNIST and fashion-MNIST datasets are grey-scale images of 28×28 pixels. The CIFAR-10 dataset consists of colored images of 32×32 pixels. We utilize a 4-layer DGP model with 100 inducing inputs per layer and a robust-max multiclass likelihood [Hernández-Lobato *et al.*, 2011].

Convolutional Skip-layer Connection (CSC): Results show that the CSC boosts the DGP performance in real world image datasets and performs best when integrated

with the NF framework.

Table 5.1: Mean test accuracy (%) achieved by NF, IPVI, SGHMC, and DSVI for 3 classification datasets with convolutions.

Dataset	MNIST		Fashion-MNIST		CIFAR-10	
	SGP	DGP 4	SGP	DGP 4	SGP	DGP 4
DSVI	97.32	99.16	86.98	91.57	47.15	75.05
SGHMC	96.41	98.15	85.84	88.14	47.32	70.78
IPVI	97.02	99.32	**87.29**	91.78	48.07	76.11
NF	**97.37**	**99.43**	87.14	**92.03**	**48.13**	**76.81**

Chapter 6

Discussion

In the previous chapters of this thesis, we have mainly discussed our contributions of developing accurate and efficient approximate inference methods for GP and DGP models. In this final chapter, we would like to look at the machine learning problem from a wider perspective. In particular, we will discuss the advantages of the GP and DGP models reviewed in this thesis and what they can offer to real-world problems which are nowadays dominated by deep learning. Before delving into that, we would like to summarize our contributions.

6.1 Summary

This thesis mainly discusses three novel approximate inference methods for GP and DGP models.

Chapter 3 presents a novel variational inference framework for a family of VBSGP regression models (e.g., VBDTC, VBFITC, VBPIC) whose approximations are variationally optimal with respect to the GP regression model enriched with various corresponding correlation structures of the observation noises. The variational Bayesian treatment of the hyperparameters enables the VBSGP regression models to

mitigate critical issues (e.g. overfitting) which plague existing variational SGP models that optimize point estimates of the hyperparameters. The stochastic variants of the VBSGP regression models can quickly achieve good predictive performance and improve their predictive performance over time, thus achieving scalability to big data. We empirically verifies the superior performance of the stochastic variant of the VBPIC over existing state-of-the-art GP models, which demonstrates its robustness against overfitting due to Bayesian model selection and capability of preserving scalability to big data through stochastic optimization.

However, VBSGP has several limitations. Firstly, VBSGP is specifically developed for the context of regression; when it comes to classification, since the data likelihood is non-Gaussian, assuming the posterior distribution to be Gaussian is restrictive. Moreover, the expressive power of VBSGP is also limited by the expressiveness of the kernel function. In order to boost the expressive power, a hierarchical composition of single-layer GPs, DGP, has been proposed. Nevertheless, unlike its single-layer counterpart, the posterior of DGP is intractable. This has motivated the development of deterministic and stochastic approximation methods for DGPs.

Unfortunately, deterministic approximation methods usually impose restrictive assumptions while stochastic approximation methods are computational costly. To this end, a novel IPVI framework is proposed that can ideally recover an unbiased posterior belief of the inducing variables and still preserve the time efficiency of VI. To achieve this, we cast the DGP inference problem as a two-player game and search for a Nash equilibrium (i.e., an unbiased posterior belief) of this game using best-response dynamics. Empirical evaluations show that IPVI outperforms the state-of-the-art approximation methods for DGPs in regression and classification tasks and accurately learns complex multi-modal posterior beliefs in our synthetic experiment and an unsupervised learning task.

Though IPVI will recover the unbiased posterior belief ideally, there is no guar-

antee the BRD algorithm converges to a Nash equilibrium in practice, this may result in suboptimal performance. Moreover, similar to the stochastic approximation methods, IPVI can only represent the posterior distribution through samples which lacks a crucial property, probability density, of a distribution. To this end, a novel and interesting variational inference framework for DGP models based on normalizing flows is introduced. To further mitigate the issue of optimization difficulty and capture the dependency of inducing variables among layers, we propose this novel normalizing flow with convolutions. The empirical experimental results reveal that the normalizing flow framework is more robust than IPVI in terms of training as well as outperforming IVPI in terms of predictive performance. This makes the NF framework a superior alternative to existing DGP methods.

6.2 Deep Gaussian Processes and Deep Learning

Deep learning [LeCun *et al.*, 2015] has been the dominating tool for a wide variety of machine learning tasks, such as image recognition, image segmentation, object detection, natural language processing, reinforcement learning and unsupervised learning. Being overshadowed by the impressive performance of deep learning, it seems that the performance deficit of DGP models has become well acknowledged which also causes the advantages of DGP to be underestimated. We take a moment to reiterate a few major issues suffered by the current deep learning methods:

- Uncertainty estimate: The uncertainty estimate is of particular importance when dealing with adversarial samples [Kurakin *et al.*, 2016]. It has been shown by [Li and Gal, 2017] that uncertainty provides additional robustness.

- Model selection: Due to their parametric nature, the desirable performance of deep learning models relies on selecting the model with the appropriate com-

plexity. In particular, for small datasets, an overly expressive model is prone to overfitting; on the other hand, when a large amount of data is available, under-fitting usually occurs if the model is not expressive enough. In contrast, the complexity of DGP models naturally grows as data is increased and thus an appropriate complexity is always maintained.

6.3 Future Directions

To further attract attentions of researchers from the deep learning community, researchers (including me) still need to resolve the challenges related with DGP to deal with real-world industrial problems. We now briefly discuss some potential solutions based on the challenges we addressed throughout this thesis.

6.3.1 Computation

Despite the advances of approximate inference methods we detailed in this thesis, Gaussian process models still have a critical computational disadvantages compared with neural networks. Gaussian process models require the inversion of the covariance matrix, which is implemented with resort to the Cholesky decomposition [De G. Matthews *et al.*, 2017]. Conversely, neural networks only require matrix multiplication which is highly robust to numerical issues and can be computed in parallel efficiently. These properties also motivated the development of modern computing hardware, GPUs and TPUs, which focus on parallel computing as well as low-precision computing. One aspect of future works on Gaussian processes is to accelerate the computation. To this end, [Wang *et al.*, 2019] developed a scalable approach which utilizes conjugate gradient method, which replaced Cholesky decomposition with Blackbox Matrix-Matrix multiplication. Another work [van der Wilk

et al., 2019] proposed a framework which directly removes the computation of the matrix inversion. However, all of these pioneer works only deal with regression tasks, a more refined framework which can generalize to other tasks (e.g. classification, dimensionality reduction) is yet to be investigated.

6.3.2 Correlated Outputs

Throughout this thesis, we only considered DGP models in which multiple output dimensions are assumed to be conditionally independent given the input to simplify computation and sampling. However, the DGP models could still be extended to model explicitly correlated outputs given the inputs. There are a few works which directly utilize the correlated outputs for single-layer Gaussian process models [Alvarez and Lawrence, 2009; Álvarez and Lawrence, 2011; Nguyen *et al.*, 2014; Zhang *et al.*, 2016]. It will be interesting to investigate how we can adapt those strategies for DGP models.

6.3.3 Optimization

Apart from the development of various deep neural network architectures which are catered to boosting the performance of deep learning, it is also undeniable that considerable effort is put on developing more effective and robust optimization techniques. For example, Dropout [Srivastava *et al.*, 2014] is proposed to reduce the issue of overfitting; Batch normalization [Ioffe and Szegedy, 2015] is proposed to reduce internal covariate shift hence making the neural network more robust to different initialization and learning rates; Adam [Kingma and Ba, 2014] is set to be the default optimizer for gradient descent in modern deep learning.

However, when it comes to Gaussian process models, more effort is put on developing practical inference methods. This raises the question of whether more refined

optimization techniques could be proposed to further improve the training in Gaussian process models. For example, [Salimbeni *et al.*, 2018] proposed to utilize natural gradient to optimize the Gaussian process models. More interestingly, [Khan *et al.*, 2018] proposed the variational Adam optimization method for Bayesian machine learning in general. This will be a valuable direction to explore in the future.

6.3.4 Deep Neural Networks and DGPs

As proved by [Neal, 1995], a single-layer fully-connected neural network with an independent and identically distributed prior over its parameters is equivalent to a Gaussian process, in the limit of infinite network width. Recently, huge amount of efforts have been put on deriving the connection between deep neural networks and Gaussian processes [Jacot *et al.*, 2018; Lee *et al.*, 2018; Matthews *et al.*, 2018; Novak *et al.*, 2019; Garriga-Alonso *et al.*, 2019]. The aim is to help the deep learning community to better understand the mechanism of neural networks and thus explain the impressive performances of deep learning models. However, it still remains an open question about how we can position deep Gaussian processes within this spectrum. I believe this will possibly broaden up our horizon and look at the deep neural networks in a different perspective.

6.3.5 Divergence Measures

Among all the approximate inference methods we proposed, KL divergence is considered to be the only measure of the distance between distributions. However, optimal transport distances such as Wasserstein distance [Villani, 2003; Peyré *et al.*, 2019] have gained substantial popularity in the generative modeling literature as they can be shown to be well-behaved in several situations where the KL divergence is either infinite or undefined [Montavon *et al.*, 2016; Arjovsky *et al.*, 2017;

Genevay *et al.*, 2018], especially when two distributions are quite different. Generally, the posterior distribution is very different from the prior distribution in the context of Gaussian processes, hence KL divergence does not work well in quantifying the difference between distributions [Gulrajani *et al.*, 2017]. It would be of great importance to adopt other divergence measures to overcome the limitations of KL divergence.

6.4 Conclusion

While there are still plenty of open challenges and problems, we hope that this thesis at least provides additional confidence and clarity for researchers who are devoting themselves to Bayesian nonparametric models, Gaussian process models in particular. It will be my honor if this thesis could offer inspirations for future works, and useful thoughts for future solutions.

Appendix A

Appendix for Chapter 2

Suppose we have two random variables $\mathbf{f}_1 \in \mathbb{R}^{D_1}$ and $\mathbf{f}_2 \in \mathbb{R}^{D_2}$ such that the joint distribution follows:

$$\begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \right)$$

where $\mathbf{K}_{21} \triangleq \mathbf{K}_{12}^\top$ denotes the covariance between $\mathbf{f}_1$ and $\mathbf{f}_2$. Then we can derive the following two interesting properties:

$$\begin{aligned} &\textbf{Marginalization} : p(\mathbf{f}_1) = \mathcal{N}(\boldsymbol{\mu}_1, \mathbf{K}_{11}), \;\; p(\mathbf{f}_2) = \mathcal{N}(\boldsymbol{\mu}_2, \mathbf{K}_{22}), \\ &\textbf{Conditioning} : p(\mathbf{f}_1|\mathbf{f}_2) = \mathcal{N}(\boldsymbol{\mu}_1 + \mathbf{K}_{12}\mathbf{K}_{22}^{-1}(\mathbf{f}_2 - \boldsymbol{\mu}_2), \mathbf{K}_{11} - \mathbf{K}_{12}\mathbf{K}_{22}^{-1}\mathbf{K}_{21}). \end{aligned} \tag{A.1}$$

Other than the properties above, suppose we have another two random variables $\mathbf{f}_a \in \mathbb{R}^{D_a}$ and $\mathbf{f}_b \in \mathbb{R}^{D_b}$ which satisfies:

$$\begin{aligned} p(\mathbf{f}_a) &= \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{K}_{aa}) \\ p(\mathbf{f}_b|\mathbf{f}_a) &= \mathcal{N}(\mathbf{M}\mathbf{f}_a + \mathbf{C}, \mathbf{K}_{bb|a}) \end{aligned}$$

then another interesting property can be obtained:

$$\textbf{Affine Transformation} : p(\mathbf{f}_b) = \mathcal{N}(\mathbf{M}\boldsymbol{\mu}_a + \mathbf{C}, \mathbf{K}_{bb|a} + \mathbf{M}\mathbf{K}_{aa}\mathbf{M}^\top). \tag{A.2}$$

Appendix B

Appendix for Chapter 3

B.1 Derivation of Log Marginal Likelihood

For all $\mathbf{f}_{\mathcal{D}}$, $\mathbf{s}_{\mathcal{I}}$, $\mathbf{\Lambda}$, and σ_f,

$$p(\mathbf{y}_{\mathcal{D}}) = \frac{p(\mathbf{y}_{\mathcal{D}}, \mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)}{p(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f | \mathbf{y}_{\mathcal{D}})} \ .$$

So,

$$\log p(\mathbf{y}_{\mathcal{D}}) = \log \frac{p(\mathbf{y}_{\mathcal{D}}, \mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)}{p(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f | \mathbf{y}_{\mathcal{D}})} \ .$$

Let $q(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)$ be an arbitrary probability density function that is independent of $\mathbf{y}_{\mathcal{D}}$. Integrating both sides of the above equation with respect to $q(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)$ yields

$$\log p(\mathbf{y}_{\mathcal{D}}) = \int q(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f) \log \frac{p(\mathbf{y}_{\mathcal{D}}, \mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)}{p(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f | \mathbf{y}_{\mathcal{D}})} \mathrm{d}\mathbf{f}_{\mathcal{D}} \mathrm{d}\mathbf{s}_{\mathcal{I}} \mathrm{d}\mathbf{\Lambda} \mathrm{d}\sigma_f \tag{B.1}$$

Using $\log(ab) = \log(a) + \log(b)$,

$$\log \frac{p(\mathbf{y}_{\mathcal{D}}, \mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)}{p(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f | \mathbf{y}_{\mathcal{D}})} = \log \frac{p(\mathbf{y}_{\mathcal{D}}, \mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)}{q(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)} + \log \frac{q(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)}{p(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f | \mathbf{y}_{\mathcal{D}})}$$

which is substituted into (B.1) to give

$$\begin{aligned}\log p(\mathbf{y}_{\mathcal{D}}) = &\int q(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f) \log \frac{p(\mathbf{y}_{\mathcal{D}}, \mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)}{q(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)} \mathrm{d}\mathbf{f}_{\mathcal{D}} \mathrm{d}\mathbf{s}_{\mathcal{I}} \mathrm{d}\mathbf{\Lambda} \mathrm{d}\sigma_f \\ &+ \int q(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f) \log \frac{q(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)}{p(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f | \mathbf{y}_{\mathcal{D}})} \mathrm{d}\mathbf{f}_{\mathcal{D}} \mathrm{d}\mathbf{s}_{\mathcal{I}} \mathrm{d}\mathbf{\Lambda} \mathrm{d}\sigma_f.\end{aligned} \tag{B.2}$$

The first and second terms on the RHS of (B.2) correspond to the variational lower bound $\mathcal{L}(q)$ and $\mathrm{KL}(q(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f) || p(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f | \mathbf{y}_{\mathcal{D}}))$, respectively.

B.2 Proof of Theorem 1

Given that

$$p(\mathbf{y}_{\mathcal{D}}, \mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f) = p(\mathbf{y}_{\mathcal{D}} | \mathbf{f}_{\mathcal{D}})\ p(\mathbf{f}_{\mathcal{D}} | \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)\ p(\mathbf{s}_{\mathcal{I}})\ p(\mathbf{\Lambda})\ p(\sigma_f)$$

$$\begin{aligned}\mathcal{L}(q) =& \int q(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f) \log \frac{p(\mathbf{y}_{\mathcal{D}}, \mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)}{q(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)} \mathrm{d}\mathbf{f}_{\mathcal{D}} \mathrm{d}\mathbf{s}_{\mathcal{I}} \mathrm{d}\mathbf{\Lambda} \mathrm{d}\sigma_f \\ =& \int q(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f) \log \frac{p(\mathbf{y}_{\mathcal{D}} | \mathbf{f}_{\mathcal{D}}) p(\mathbf{f}_{\mathcal{D}} | \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f) p(\mathbf{s}_{\mathcal{I}}) p(\mathbf{\Lambda}) p(\sigma_f)}{p(\mathbf{f}_{\mathcal{D}} | \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f) q(\mathbf{s}_{\mathcal{I}}) q(\mathbf{\Lambda}) q(\sigma_f)} \mathrm{d}\mathbf{f}_{\mathcal{D}} \mathrm{d}\mathbf{s}_{\mathcal{I}} \mathrm{d}\mathbf{\Lambda} \mathrm{d}\sigma_f \\ =& \int q(\mathbf{f}_{\mathcal{D}}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f) \log \frac{p(\mathbf{y}_{\mathcal{D}} | \mathbf{f}_{\mathcal{D}}) p(\mathbf{s}_{\mathcal{I}}) p(\mathbf{\Lambda}) p(\sigma_f)}{q(\mathbf{s}_{\mathcal{I}}) q(\mathbf{\Lambda}) q(\sigma_f)}\ \mathrm{d}\mathbf{f}_{\mathcal{D}}\ \mathrm{d}\mathbf{s}_{\mathcal{I}}\ \mathrm{d}\mathbf{\Lambda}\ \mathrm{d}\sigma_f \\ =& \int p(\mathbf{f}_{\mathcal{D}} | \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)\ q(\mathbf{s}_{\mathcal{I}})\ q(\mathbf{\Lambda})\ q(\sigma_f) \Bigg(\log p(\mathbf{y}_{\mathcal{D}} | \mathbf{f}_{\mathcal{D}}) + \log \frac{p(\mathbf{s}_{\mathcal{I}})}{q(\mathbf{s}_{\mathcal{I}})} \\ &+ \log \frac{p(\mathbf{\Lambda})}{q(\mathbf{\Lambda})} + \log \frac{p(\sigma_f)}{q(\sigma_f)} \Bigg) \mathrm{d}\mathbf{f}_{\mathcal{D}}\ \mathrm{d}\mathbf{s}_{\mathcal{I}}\ \mathrm{d}\mathbf{\Lambda}\ \mathrm{d}\sigma_f \\ =& \mathcal{F}(q) + \int q(\mathbf{s}_{\mathcal{I}}) \log \frac{p(\mathbf{s}_{\mathcal{I}})}{q(\mathbf{s}_{\mathcal{I}})}\ \mathrm{d}\mathbf{s}_{\mathcal{I}} + \int q(\mathbf{\Lambda}) \log \frac{p(\mathbf{\Lambda})}{q(\mathbf{\Lambda})}\ \mathrm{d}\mathbf{\Lambda}\ + \int q(\sigma_f) \log \frac{p(\sigma_f)}{q(\sigma_f)}\ \mathrm{d}\sigma_f\end{aligned}$$

where

$$\begin{aligned}\mathcal{F}(q) &= \int q(\mathbf{s}_\mathcal{I})\ \mathcal{G}(q, \mathbf{s}_\mathcal{I})\ \mathrm{d}\mathbf{s}_\mathcal{I} \\ \mathcal{G}(q, \mathbf{s}_\mathcal{I}) &= \int q(\sigma_f)\ q(\mathbf{\Lambda})\ \mathcal{H}(\mathbf{s}_\mathcal{I}, \mathbf{\Lambda}, \sigma_f)\ \mathrm{d}\mathbf{\Lambda}\ \mathrm{d}\sigma_f \\ \mathcal{H}(\mathbf{s}_\mathcal{I}, \mathbf{\Lambda}, \sigma_f) &= \int p(\mathbf{f}_\mathcal{D}|\mathbf{s}_\mathcal{I}, \mathbf{\Lambda}, \sigma_f)\ \log p(\mathbf{y}_\mathcal{D}|\mathbf{f}_\mathcal{D})\ \mathrm{d}\mathbf{f}_\mathcal{D}\ .\end{aligned}$$

Let us first derive the closed-form expression of $H(\mathbf{s}_\mathcal{I}, \mathbf{\Lambda}, \sigma_f)$:

$$\begin{aligned}&H(\mathbf{s}_\mathcal{I}, \mathbf{\Lambda}, \sigma_f) = \int p(\mathbf{f}_\mathcal{D}|\mathbf{s}_\mathcal{I}, \mathbf{\Lambda}, \sigma_f) \log p(\mathbf{y}_\mathcal{D}|\mathbf{f}_\mathcal{D})\ \mathrm{d}\mathbf{f}_\mathcal{D} \\ &= \int p(\mathbf{f}_\mathcal{D}|\mathbf{s}_\mathcal{I}, \mathbf{\Lambda}, \sigma_f) \left(-\frac{|\mathcal{D}|}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{C}_{\mathcal{D}\mathcal{D}}| - \frac{1}{2}(\mathbf{y}_\mathcal{D} - \mathbf{f}_\mathcal{D})^\top \mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}(\mathbf{y}_\mathcal{D} - \mathbf{f}_\mathcal{D}) \right) \mathrm{d}\mathbf{f}_\mathcal{D} \\ &= -\frac{|\mathcal{D}|}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{C}_{\mathcal{D}\mathcal{D}}| - \mathbb{E}_{p(\mathbf{f}_\mathcal{D}|\mathbf{s}_\mathcal{I}, \mathbf{\Lambda}, \sigma_f)}\left[\frac{1}{2}(\mathbf{y}_\mathcal{D} - \mathbf{f}_\mathcal{D})^\top \mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}(\mathbf{y}_\mathcal{D} - \mathbf{f}_\mathcal{D})\right] \\ &= -\frac{|\mathcal{D}|}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{C}_{\mathcal{D}\mathcal{D}}| - \frac{1}{2}(\mathbf{y}_\mathcal{D} - \mathbf{K}_{\mathcal{D}\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{s}_\mathcal{I})^\top \mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}(\mathbf{y}_\mathcal{D} - \mathbf{K}_{\mathcal{D}\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{s}_\mathcal{I}) \\ &\quad -\frac{1}{2}\mathrm{Tr}[\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{K}_{\mathcal{D}\mathcal{D}}] + \frac{1}{2}\mathrm{Tr}[\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{K}_{\mathcal{D}\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{K}_{\mathcal{I}\mathcal{D}}]\end{aligned}$$

where the last equality follows from eq. 380 of [Petersen *et al.*, 2012] and (3.11).

The closed-form expression of $\mathcal{G}(q, \mathbf{s}_\mathcal{I})$ can then be derived as follows:

$$\begin{aligned}&\mathcal{G}(q, \mathbf{s}_\mathcal{I}) = \int q(\sigma_f)\ q(\mathbf{\Lambda})\ \mathcal{H}(\mathbf{s}_\mathcal{I}, \mathbf{\Lambda}, \sigma_f)\ \mathrm{d}\mathbf{\Lambda}\ \mathrm{d}\sigma_f \\ &= -\frac{|\mathcal{D}|}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{C}_{\mathcal{D}\mathcal{D}}| - \frac{1}{2}\mathbb{E}_{q(\mathbf{\Lambda}, \sigma_f)}\left[(\mathbf{y}_\mathcal{D} - \mathbf{K}_{\mathcal{D}\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{s}_\mathcal{I})^\top \mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}(\mathbf{y}_\mathcal{D} - \mathbf{K}_{\mathcal{D}\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{s}_\mathcal{I})\right] \\ &\quad -\frac{1}{2}\mathrm{Tr}[\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbb{E}_{q(\mathbf{\Lambda}, \sigma_f)}[\mathbf{K}_{\mathcal{D}\mathcal{D}}]] + \frac{1}{2}\mathrm{Tr}[\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbb{E}_{q(\mathbf{\Lambda}, \sigma_f)}[\mathbf{K}_{\mathcal{I}\mathcal{D}}\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{K}_{\mathcal{D}\mathcal{I}}]] \\ &= -\frac{|\mathcal{D}|}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{C}_{\mathcal{D}\mathcal{D}}| - \frac{1}{2}\mathbf{y}_\mathcal{D}^\top \mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{y}_\mathcal{D} + \mathbf{s}_\mathcal{I}^\top \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Omega}_{\mathcal{I}\mathcal{D}}\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{y}_\mathcal{D} - \frac{1}{2}\mathbf{s}_\mathcal{I}^\top \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Psi}_{\mathcal{I}\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{s}_\mathcal{I} \\ &\quad -\frac{1}{2}\mathrm{Tr}[\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{\Upsilon}_{\mathcal{D}\mathcal{D}}] + \frac{1}{2}\mathrm{Tr}[\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Psi}_{\mathcal{I}\mathcal{I}}]\end{aligned}$$

such that the last equality follows from

$$\begin{aligned}
&\mathbb{E}_{q(\boldsymbol{\Lambda},\sigma_f)}\left[(\mathbf{y}_\mathcal{D} - \mathbf{K}_{\mathcal{D}\mathcal{I}}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{s}_\mathcal{I})^\top \mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}(\mathbf{y}_\mathcal{D} - \mathbf{K}_{\mathcal{D}\mathcal{I}}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{s}_\mathcal{I})\right] \\
&= \mathbb{E}_{q(\boldsymbol{\Lambda},\sigma_f)}\left[(\mathbf{y}_\mathcal{D}^\top - \mathbf{s}_\mathcal{I}^\top\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{K}_{\mathcal{I}\mathcal{D}})\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}(\mathbf{y}_\mathcal{D} - \mathbf{K}_{\mathcal{D}\mathcal{I}}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{s}_\mathcal{I})\right] \\
&= \mathbf{y}_\mathcal{D}^\top\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{y}_\mathcal{D} - 2\mathbf{s}_\mathcal{I}^\top\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}}\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{y}_\mathcal{D} + \mathbf{s}_\mathcal{I}^\top\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{s}_\mathcal{I}.
\end{aligned}$$

The closed-form expression of $\mathcal{F}(q)$ is

$$\mathcal{F}(q) = \int q(\mathbf{s}_\mathcal{I})\ \mathcal{G}(q, \mathbf{s}_\mathcal{I})\ \mathrm{d}\mathbf{s}_\mathcal{I} = \mathbb{E}_{q(\mathbf{s}_\mathcal{I})}[\mathcal{G}(q, \mathbf{s}_\mathcal{I})]$$

where, using $q(\mathbf{s}_\mathcal{I}) = \mathcal{N}(\mathbf{m}, \mathbf{S})$,

$$\begin{aligned}
\mathbb{E}_{q(\mathbf{s}_\mathcal{I})}[\mathcal{G}(q, \mathbf{s}_\mathcal{I})] = &-\frac{|\mathcal{D}|}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{C}_{\mathcal{D}\mathcal{D}}| - \frac{1}{2}\mathbf{y}_\mathcal{D}^\top\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{y}_\mathcal{D} + \mathbf{m}^\top\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}}\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{y}_\mathcal{D} \\
&-\frac{1}{2}\mathbf{m}^\top\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m} - \frac{1}{2}\mathrm{Tr}[\mathbf{S}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}] - \frac{1}{2}\mathrm{Tr}[\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\boldsymbol{\Upsilon}_{\mathcal{D}\mathcal{D}}] \\
&+\frac{1}{2}\mathrm{Tr}[\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}]
\end{aligned}$$

such that $\mathbb{E}_{q(\mathbf{s}_\mathcal{I})}[\mathcal{G}(q, \mathbf{s}_\mathcal{I})]$ is derived using eqs. 374 and 380 of [Petersen *et al.*, 2012]. Since

$$\int q(\mathbf{s}_\mathcal{I})\log\frac{p(\mathbf{s}_\mathcal{I})}{q(\mathbf{s}_\mathcal{I})}\ \mathrm{d}\mathbf{s}_\mathcal{I} = \mathbb{E}_{q(\mathbf{s}_\mathcal{I})}[\log p(\mathbf{s}_\mathcal{I})] + \mathbb{H}[q(\mathbf{s}_\mathcal{I})]$$

where

$$\mathbb{E}_{q(\mathbf{s}_\mathcal{I})}[\log p(\mathbf{s}_\mathcal{I})] = -\frac{|\mathcal{I}|}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}| - \frac{1}{2}\mathbf{m}^\top\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m} - \frac{1}{2}\mathrm{Tr}[\mathbf{S}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}]$$

and

$$\mathbb{H}[q(\mathbf{s}_\mathcal{I})] = \frac{|\mathcal{I}|}{2}\log 2\pi + \frac{|\mathcal{I}|}{2} + \frac{1}{2}\log|\mathbf{S}|$$

denotes a Gaussian entropy with respect to $q(\mathbf{s}_\mathcal{I})$,

$$\int q(\boldsymbol{\Lambda})\log\frac{p(\boldsymbol{\Lambda})}{q(\boldsymbol{\Lambda})}d\boldsymbol{\Lambda} = -\frac{1}{2}\boldsymbol{\nu}^\top\boldsymbol{\nu} - \frac{1}{2}\mathrm{Tr}[\boldsymbol{\Xi}] + \frac{1}{2}\log|\boldsymbol{\Xi}| + \frac{d}{2}\ ,$$

and

$$\int q(\sigma_f) \log \frac{p(\sigma_f)}{q(\sigma_f)} \, \mathrm{d}\sigma_f = -\frac{1}{2}\alpha^2 - \frac{1}{2}\beta + \frac{1}{2}\log\beta + \frac{1}{2} \, ,$$

$$\mathcal{L}(q) = \mathcal{F}(q) + \int q(\mathbf{s}_\mathcal{I}) \log \frac{p(\mathbf{s}_\mathcal{I})}{q(\mathbf{s}_\mathcal{I})} \, \mathrm{d}\mathbf{s}_\mathcal{I} + \int q(\mathbf{\Lambda}) \log \frac{p(\mathbf{\Lambda})}{q(\mathbf{\Lambda})} \, \mathrm{d}\mathbf{\Lambda} + \int q(\sigma_f) \log \frac{p(\sigma_f)}{q(\sigma_f)} \, \mathrm{d}\sigma_f$$

$$=$$

$$-\frac{|\mathcal{D}|}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{C}_{\mathcal{D}\mathcal{D}}| - \frac{1}{2}\mathbf{y}_\mathcal{D}^\top \mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{y}_\mathcal{D} + \mathbf{m}^\top \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Omega}_{\mathcal{I}\mathcal{D}}\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{y}_\mathcal{D} - \frac{1}{2}\mathbf{m}^\top(\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Psi}_{\mathcal{I}\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1} + \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1})\mathbf{m}$$
$$-\frac{1}{2}\mathrm{Tr}[\mathbf{S}(\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Psi}_{\mathcal{I}\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1} + \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1})] - \frac{1}{2}\mathrm{Tr}[\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{\Upsilon}_{\mathcal{D}\mathcal{D}}] + \frac{1}{2}\mathrm{Tr}[\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Psi}_{\mathcal{I}\mathcal{I}}] - \frac{1}{2}\log|\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}| + \frac{|\mathcal{I}|}{2} + \frac{1}{2}\log|\mathbf{S}|$$
$$-\frac{1}{2}\boldsymbol{\nu}^\top\boldsymbol{\nu} - \frac{1}{2}\mathrm{Tr}[\mathbf{\Xi}] + \frac{1}{2}\log|\mathbf{\Xi}| + \frac{d}{2} - \frac{1}{2}\alpha^2 - \frac{1}{2}\beta + \frac{1}{2}\log\beta + \frac{1}{2}$$

$$=$$

$$\frac{1}{2}\Big(2\mathbf{m}^\top\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Omega}_{\mathcal{I}\mathcal{D}}\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{y}_\mathcal{D} - \mathbf{m}^\top(\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Psi}_{\mathcal{I}\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1} + \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1})\mathbf{m} - \mathrm{Tr}[\mathbf{S}(\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Psi}_{\mathcal{I}\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1} + \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1})]$$
$$-\mathrm{Tr}[\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{\Upsilon}_{\mathcal{D}\mathcal{D}}] + \mathrm{Tr}[\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Psi}_{\mathcal{I}\mathcal{I}}] + \log|\mathbf{S}| - \boldsymbol{\nu}^\top\boldsymbol{\nu} - \mathrm{Tr}[\mathbf{\Xi}] + \log|\mathbf{\Xi}| - \alpha^2 - \beta + \log\beta\Big) + \mathrm{const}$$

where const absorbs all terms independent of $\mathbf{m}$, $\mathbf{S}$, $\boldsymbol{\nu}$, $\mathbf{\Xi}, \alpha, \beta$. Then, by setting

$$\frac{\partial\mathcal{L}}{\partial\mathbf{m}} = \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Omega}_{\mathcal{I}\mathcal{D}}\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{y}_\mathcal{D} - (\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Psi}_{\mathcal{I}\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1} + \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1})\mathbf{m} \, ,$$
$$\frac{\partial\mathcal{L}}{\partial\mathbf{S}} = \frac{1}{2}\mathbf{S}^{-1} - \frac{1}{2}(\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Psi}_{\mathcal{I}\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1} + \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1})$$

to zero, it can be derived that $\mathcal{L}(q)$ is maximized at $q^*(\mathbf{s}_\mathcal{I}) = \mathcal{N}(\mathbf{m}^*, \mathbf{S}^*)$ where

$$\begin{aligned} \mathbf{m}^* &= (\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Psi}_{\mathcal{I}\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1} + \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1})^{-1}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Omega}_{\mathcal{I}\mathcal{D}}\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{y}_\mathcal{D} \, , \\ \mathbf{S}^* &= (\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Psi}_{\mathcal{I}\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1} + \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1})^{-1} \, . \end{aligned} \tag{B.3}$$

By substituting

$$(\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Psi}_{\mathcal{I}\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1} + \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1})^{-1} = ((\mathbf{I} + \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Psi}_{\mathcal{I}\mathcal{I}})\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1})^{-1} = \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}(\mathbf{I} + \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{\Psi}_{\mathcal{I}\mathcal{I}})^{-1}$$
$$= \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}(\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}(\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}} + \mathbf{\Psi}_{\mathcal{I}\mathcal{I}}))^{-1} = \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}(\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}} + \mathbf{\Psi}_{\mathcal{I}\mathcal{I}})^{-1}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}$$

into (B.3), (3.15) in Theorem 1 results. Using (3.15),

$$\mathbf{m}^{*\top}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}}\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{y}_{\mathcal{D}} = \mathbf{y}_{\mathcal{D}}^{\top}\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}}^{\top}(\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}+\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}})^{-1}\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}}\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{y}_{\mathcal{D}} \ ,$$
$$\mathbf{m}^{*\top}\Big(\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}+\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\Big)\mathbf{m}^{*} = \mathbf{y}_{\mathcal{D}}^{\top}\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}}^{\top}(\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}+\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}})^{-1}\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}}\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{y}_{\mathcal{D}} \ ,$$
$$\mathrm{Tr}(\mathbf{S}^{*}(\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}+\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1})) = |\mathcal{I}|$$

which reduce $\mathcal{L}(q)$ to

$$\mathcal{L}(q) = \frac{1}{2}\Big(\mathbf{y}_{\mathcal{D}}^{\top}\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}}^{\top}(\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}+\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}})^{-1}\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}}\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{y}_{\mathcal{D}} - \mathrm{Tr}[\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\boldsymbol{\Upsilon}_{\mathcal{D}\mathcal{D}}] + \mathrm{Tr}[\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}]$$
$$- \log|\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}+\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}| - \boldsymbol{\nu}^{\top}\boldsymbol{\nu} - \mathrm{Tr}[\boldsymbol{\Xi}] + \log|\boldsymbol{\Xi}| - \alpha^{2} - \beta + \log\beta\Big) + \mathrm{const} \ . \tag{B.4}$$

B.3 Proof of Theorem 2

Let

$$\begin{aligned}
&\frac{\partial\widetilde{\mathcal{L}}}{\partial\mathbf{m}} \triangleq \frac{B}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\frac{\partial\mathcal{L}_s}{\partial\mathbf{m}} - \boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m} \ , \frac{\partial\widetilde{\mathcal{L}}}{\partial\boldsymbol{\Xi}} \triangleq \frac{B}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\frac{\partial\mathcal{L}_s}{\partial\boldsymbol{\Xi}} - \frac{1}{2}\mathbf{I} + \frac{1}{2}\boldsymbol{\Xi}^{-1}, \\
&\frac{\partial\widetilde{\mathcal{L}}}{\partial\mathbf{S}} \triangleq \frac{B}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\frac{\partial\mathcal{L}_s}{\partial\mathbf{S}} + \frac{1}{2}\mathbf{S}^{-1} - \frac{1}{2}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1} \ , \ \frac{\partial\widetilde{\mathcal{L}}}{\partial\boldsymbol{\nu}} \triangleq \frac{B}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\frac{\partial\mathcal{L}_s}{\partial\boldsymbol{\nu}} - \boldsymbol{\nu}, \\
&\frac{\partial\widetilde{\mathcal{L}}}{\partial\alpha} \triangleq \frac{B}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\frac{\partial\mathcal{L}_s}{\partial\alpha} - \alpha \ , \ \frac{\partial\widetilde{\mathcal{L}}}{\partial\beta} \triangleq \frac{B}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\frac{\partial\mathcal{L}_s}{\partial\beta} - \frac{\beta-1}{2\beta}
\end{aligned} \tag{B.5}$$

where

$$
\begin{aligned}
\frac{\partial \mathcal{L}_s}{\partial \mathbf{m}} &= \boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}_s}\mathbf{C}_{\mathcal{D}_s\mathcal{D}_s}^{-1}\mathbf{y}_{\mathcal{D}_s} - \boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m} \ , \\
\frac{\partial \mathcal{L}_s}{\partial \mathbf{S}} &= -\frac{1}{2}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1} \ , \\
\frac{\partial \mathcal{L}_s}{\partial \boldsymbol{\nu}} &= \mathbf{m}^\top\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\frac{\partial \boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}_s}}{\partial \boldsymbol{\nu}}\mathbf{C}_{\mathcal{D}_s\mathcal{D}_s}^{-1}\mathbf{y}_{\mathcal{D}_s} - \frac{1}{2}\mathbf{m}^\top\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\frac{\partial \boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}}{\partial \boldsymbol{\nu}}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m} \\
&\quad -\frac{1}{2}\mathrm{Tr}\left[\mathbf{S}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\frac{\partial \boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}}{\partial \boldsymbol{\nu}}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\right] - \frac{1}{2}\mathrm{Tr}\left[\mathbf{C}_{\mathcal{D}_s\mathcal{D}_s}^{-1}\frac{\partial \boldsymbol{\Upsilon}_{\mathcal{D}_s\mathcal{D}_s}}{\partial \boldsymbol{\nu}}\right] \\
&\quad +\frac{1}{2}\mathrm{Tr}\left[\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\frac{\partial \boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}}{\partial \boldsymbol{\nu}}\right] \ , \\
\frac{\partial \mathcal{L}_s}{\partial \boldsymbol{\Xi}} &= \mathbf{m}^\top\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\frac{\partial \boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}_s}}{\partial \boldsymbol{\Xi}}\mathbf{C}_{\mathcal{D}_s\mathcal{D}_s}^{-1}\mathbf{y}_{\mathcal{D}_s} - \frac{1}{2}\mathbf{m}^\top\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\frac{\partial \boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}}{\partial \boldsymbol{\Xi}}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m} \\
&\quad -\frac{1}{2}\mathrm{Tr}\left[\mathbf{S}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\frac{\partial \boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}}{\partial \boldsymbol{\Xi}}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\right] - \frac{1}{2}\mathrm{Tr}\left[\mathbf{C}_{\mathcal{D}_s\mathcal{D}_s}^{-1}\frac{\partial \boldsymbol{\Upsilon}_{\mathcal{D}_s\mathcal{D}_s}}{\partial \boldsymbol{\Xi}}\right] \\
&\quad +\frac{1}{2}\mathrm{Tr}\left[\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\frac{\partial \boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}}{\partial \boldsymbol{\Xi}}\right] \ , \\
\frac{\partial \mathcal{L}_s}{\partial \alpha} &= \mathbf{m}^\top\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\frac{\partial \boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}_s}}{\partial \alpha}\mathbf{C}_{\mathcal{D}_s\mathcal{D}_s}^{-1}\mathbf{y}_{\mathcal{D}_s} - \frac{1}{2}\mathbf{m}^\top\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\frac{\partial \boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}}{\partial \alpha}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m} \\
&\quad -\frac{1}{2}\mathrm{Tr}\left[\mathbf{S}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\frac{\partial \boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}}{\partial \alpha}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\right] - \frac{1}{2}\mathrm{Tr}\left[\mathbf{C}_{\mathcal{D}_s\mathcal{D}_s}^{-1}\frac{\partial \boldsymbol{\Upsilon}_{\mathcal{D}_s\mathcal{D}_s}}{\partial \alpha}\right] \\
&\quad +\frac{1}{2}\mathrm{Tr}\left[\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\frac{\partial \boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}}{\partial \alpha}\right] \ , \\
\frac{\partial \mathcal{L}_s}{\partial \beta} &= \mathbf{m}^\top\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\frac{\partial \boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}_s}}{\partial \beta}\mathbf{C}_{\mathcal{D}_s\mathcal{D}_s}^{-1}\mathbf{y}_{\mathcal{D}_s} - \frac{1}{2}\mathbf{m}^\top\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\frac{\partial \boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}}{\partial \beta}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m} \\
&\quad -\frac{1}{2}\mathrm{Tr}\left[\mathbf{S}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\frac{\partial \boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}}{\partial \beta}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\right] - \frac{1}{2}\mathrm{Tr}\left[\mathbf{C}_{\mathcal{D}_s\mathcal{D}_s}^{-1}\frac{\partial \boldsymbol{\Upsilon}_{\mathcal{D}_s\mathcal{D}_s}}{\partial \beta}\right] \\
&\quad +\frac{1}{2}\mathrm{Tr}\left[\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\frac{\partial \boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}}{\partial \beta}\right] \ ,
\end{aligned}
$$

and the closed-form expressions of $\partial\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}_s}/\partial\boldsymbol{\nu}$, $\partial\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}/\partial\boldsymbol{\nu}$, $\partial\boldsymbol{\Upsilon}_{\mathcal{D}_s\mathcal{D}_s}/\partial\boldsymbol{\nu}$, $\partial\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}_s}/\partial\boldsymbol{\Xi}$, $\partial\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}/\partial\boldsymbol{\Xi}$, $\partial\boldsymbol{\Upsilon}_{\mathcal{D}_s\mathcal{D}_s}/\partial\boldsymbol{\Xi}$ $\partial\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}_s}/\partial\alpha$, $\partial\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}/\partial\alpha$, $\partial\boldsymbol{\Upsilon}_{\mathcal{D}_s\mathcal{D}_s}/\partial\alpha$, $\partial\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}_s}/\partial\beta$, $\partial\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}/\partial\beta$, and $\partial\boldsymbol{\Upsilon}_{\mathcal{D}_s\mathcal{D}_s}/\partial\beta$ are given in Appendix B.4.2.

Then, since

$$
\begin{aligned}
&\mathbb{E}[\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}_s}\mathbf{C}_{\mathcal{D}_s\mathcal{D}_s}^{-1}\mathbf{y}_{\mathcal{D}_s} - \boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m}] \\
&= \sum_{i=1}^{B} p(s=i)(\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}_i}\mathbf{C}_{\mathcal{D}_i\mathcal{D}_i}^{-1}\mathbf{y}_{\mathcal{D}_i} - \boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{i}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m}) \\
&= \sum_{i=1}^{B} \frac{1}{B}(\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}_i}\mathbf{C}_{\mathcal{D}_i\mathcal{D}_i}^{-1}\mathbf{y}_{\mathcal{D}_i} - \boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{i}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m}) \\
&= \frac{1}{B}\sum_{i=1}^{B} \boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}_i}\mathbf{C}_{\mathcal{D}_i\mathcal{D}_i}^{-1}\mathbf{y}_{\mathcal{D}_i} - \boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{i}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m} \ ,
\end{aligned}
$$

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{s\in\mathcal{S}} \boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}_s}\mathbf{C}_{\mathcal{D}_s\mathcal{D}_s}^{-1}\mathbf{y}_{\mathcal{D}_s} - \boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{s}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m}\right] \\
&= \frac{|\mathcal{S}|}{B}\sum_{i=1}^{B} \boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Omega}_{\mathcal{I}\mathcal{D}_i}\mathbf{C}_{\mathcal{D}_i\mathcal{D}_i}^{-1}\mathbf{y}_{\mathcal{D}_i} - \boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\boldsymbol{\Psi}_{\mathcal{I}\mathcal{I}}^{i}\boldsymbol{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m} \ .
\end{aligned}
$$

It follows that $\mathbb{E}[\partial\widetilde{\mathcal{L}}/\partial\mathbf{m}] = \partial\mathcal{L}/\partial\mathbf{m}$. The proofs for $\mathbb{E}[\partial\widetilde{\mathcal{L}}/\partial\mathbf{S}] = \partial\mathcal{L}/\partial\mathbf{S}$, $\mathbb{E}[\partial\widetilde{\mathcal{L}}/\partial\boldsymbol{\nu}] = \partial\mathcal{L}/\partial\boldsymbol{\nu}$, $\mathbb{E}[\partial\widetilde{\mathcal{L}}/\partial\boldsymbol{\Xi}] = \partial\mathcal{L}/\partial\boldsymbol{\Xi}$, $\mathbb{E}[\partial\widetilde{\mathcal{L}}/\partial\alpha] = \partial\mathcal{L}/\partial\alpha$, and $\mathbb{E}[\partial\widetilde{\mathcal{L}}/\partial\beta] = \partial\mathcal{L}/\partial\beta$ follow a similar procedure as the above.

B.4 Expectation and Derivatives

B.4.1 Expectation

Let $\mathbf{\Omega}_{\mathcal{ID}} \triangleq (\omega_{\mathbf{zx}})_{\mathbf{z}\in\mathcal{I},\mathbf{x}\in\mathcal{D}}$, $\mathbf{z} \triangleq (z_1,\ldots,z_d)^\top$, and $\mathbf{x} \triangleq (x_1,\ldots,x_d)^\top$. Since $\mathbf{\Omega}_{\mathcal{ID}} \triangleq \mathbb{E}_{q(\mathbf{\Lambda},\sigma_f)}(\mathbf{K}_{\mathcal{ID}})$,

$$
\begin{aligned}
\omega_{\mathbf{zx}} &= \int q(\sigma_f)\; q(\mathbf{\Lambda})\; \mathrm{cov}[s_{\mathbf{z}}, f_{\mathbf{x}}]\; \mathrm{d}\mathbf{\Lambda}\; \mathrm{d}\sigma_f \\
&= \int q(\sigma_f) \left(\int q(\mathbf{\Lambda})\; \sigma_f \exp\!\left(-\frac{1}{2}\sum_{k=1}^{d}(\lambda_k x_k - z_k)^2\right)\mathrm{d}\mathbf{\Lambda}\right)\mathrm{d}\sigma_f \\
&= \int q(\sigma_f)\; \sigma_f \prod_{k=1}^{d} \int \exp\!\left(-\frac{1}{2}\sum_{k=1}^{d}(\lambda_k x_k - z_k)^2\right)\mathcal{N}(\lambda_k|\nu_k,\xi_k)\; \mathrm{d}\lambda_k\; \mathrm{d}\sigma_f \\
&= \int q(\sigma_f)\; \sigma_f \prod_{k=1}^{d} (\xi_k x_k^2 + 1)^{-\frac{1}{2}} \exp\left(-\frac{(x_k\nu_k - z_k)^2}{2(\xi_k x_k^2 + 1)}\right)\mathrm{d}\sigma_f \\
&= \alpha \prod_{k=1}^{d} (\xi_k x_k^2 + 1)^{-\frac{1}{2}} \exp\left(-\frac{(x_k\nu_k - z_k)^2}{2(\xi_k x_k^2 + 1)}\right).
\end{aligned}
$$

Since $\mathbf{C}_{\mathcal{DD}}$ is a block-diagonal matrix constructed using the B blocks $\mathbf{C}_{\mathcal{D}_i\mathcal{D}_i}$ for $i = 1,\ldots,B$, $\mathbf{C}^{-1}_{\mathcal{DD}}$ is also a block-diagonal matrix constructed using the B blocks $\mathbf{C}^{-1}_{\mathcal{D}_i\mathcal{D}_i}$ for $i = 1,\ldots,B$. Let $\mathbf{C}^{-1}_{\mathcal{D}_i\mathcal{D}_i} \triangleq (c^i_{\mathbf{xx}'})_{\mathbf{x},\mathbf{x}'\in\mathcal{D}_i}$. Let $\mathbf{\Psi}_{\mathcal{II}} \triangleq (\psi_{\mathbf{zz}'})_{\mathbf{z},\mathbf{z}'\in\mathcal{I}}$, $\mathbf{z}' \triangleq (z'_1,\ldots,z'_d)^\top$, and $\mathbf{x}' \triangleq (x'_1,\ldots,x'_d)^\top$. Since $\mathbf{\Psi}_{\mathcal{II}} \triangleq \mathbb{E}_{q(\mathbf{\Lambda},\sigma_f)}(\mathbf{K}_{\mathcal{ID}}\mathbf{C}^{-1}_{\mathcal{DD}}\mathbf{K}_{\mathcal{DI}})$,

$$
\begin{aligned}
\psi_{\mathbf{z}\mathbf{z}'} &= \int q(\sigma_f) \sum_{i=1}^{B} \sum_{\mathbf{x},\mathbf{x}' \in \mathcal{D}_i} \mathbb{E}_{\mathbf{\Lambda}} \left[\mathrm{cov}[s_{\mathbf{z}}, f_{\mathbf{x}}] \; c^i_{\mathbf{x}\mathbf{x}'} \; \mathrm{cov}[f_{\mathbf{x}'}, s_{\mathbf{z}'}] \right] \mathrm{d}\sigma_f \\
&= \int q(\sigma_f) \sum_{i=1}^{B} \sum_{\mathbf{x},\mathbf{x}' \in \mathcal{D}_i} c^i_{\mathbf{x}\mathbf{x}'} \; \mathbb{E}_{\mathbf{\Lambda}} \left[\mathrm{cov}[s_{\mathbf{z}}, f_{\mathbf{x}}] \; \mathrm{cov}[f_{\mathbf{x}'}, s_{\mathbf{z}'}] \right] \mathrm{d}\sigma_f \\
&= \int q(\sigma_f) \sum_{i=1}^{B} \sum_{\mathbf{x},\mathbf{x}' \in \mathcal{D}_i} \sigma_f^2 \; c^i_{\mathbf{x}\mathbf{x}'} \prod_{k=1}^{d} \Bigg\{ (\xi_k(x_k^2 + x_k'^2) + 1)^{-\frac{1}{2}} \\
&\qquad \exp\left(-\frac{\xi_k(z_k' x_k - z_k x_k')^2 + (x_k \nu_k - z_k)^2 + (x_k' \nu_k - z_k')^2}{2(\xi_k(x_k^2 + x_k'^2) + 1)} \right) \Bigg\} \, \mathrm{d}\sigma_f \\
&= \sum_{i=1}^{B} \sum_{\mathbf{x},\mathbf{x}' \in \mathcal{D}_i} (\beta + \alpha^2) \; c^i_{\mathbf{x}\mathbf{x}'} \prod_{k=1}^{d} \Bigg\{ (\xi_k(x_k^2 + x_k'^2) + 1)^{-\frac{1}{2}} \\
&\qquad \exp\left(-\frac{\xi_k(z_k' x_k - z_k x_k')^2 + (x_k \nu_k - z_k)^2 + (x_k' \nu_k - z_k')^2}{2(\xi_k(x_k^2 + x_k'^2) + 1)} \right) \Bigg\} .
\end{aligned}
$$

Let $\mathbf{\Upsilon}_{\mathcal{D}\mathcal{D}} \triangleq (\gamma_{\mathbf{x}\mathbf{x}'})_{\mathbf{x},\mathbf{x}' \in \mathcal{D}}$. Since $\mathbf{\Upsilon}_{\mathcal{D}\mathcal{D}} \triangleq \mathbb{E}_{q(\mathbf{\Lambda},\sigma_f)}(\mathbf{K}_{\mathcal{D}\mathcal{D}})$,

$$
\begin{aligned}
\gamma_{\mathbf{x}\mathbf{x}'} &= \int q(\sigma_f) q(\mathbf{\Lambda}) \; k_{\mathbf{x}\mathbf{x}'} \; \mathrm{d}\mathbf{\Lambda} \; \mathrm{d}\sigma_f \\
&= \int q(\sigma_f) \int q(\mathbf{\Lambda}) \sigma_f^2 \exp\left(-\frac{1}{2} \sum_{k=1}^{d} \lambda_k^2 (x_k - x_k')^2 \right) \mathrm{d}\mathbf{\Lambda} \mathrm{d}\sigma_f \\
&= \int q(\sigma_f) \; \sigma_f^2 \prod_{k=1}^{d} \int \exp\left(-\frac{1}{2} \sum_{k=1}^{d} \lambda_k^2 (x_k - x_k')^2 \right) \mathcal{N}(\lambda_k | \nu_k, \xi_k) \; \mathrm{d}\lambda_k \; \mathrm{d}\sigma_f \\
&= \int q(\sigma_f) \; \sigma_f^2 \prod_{k=1}^{d} (\xi_k (x_k - x_k')^2 + 1)^{-\frac{1}{2}} \exp\left(-\frac{\nu_k^2 (x_k - x_k')^2}{2(\xi_k (x_k - x_k')^2 + 1)} \right) \mathrm{d}\sigma_f \\
&= (\beta + \alpha^2) \prod_{k=1}^{d} (\xi_k (x_k - x_k')^2 + 1)^{-\frac{1}{2}} \exp\left(-\frac{\nu_k^2 (x_k - x_k')^2}{2(\xi_k (x_k - x_k')^2 + 1)} \right) .
\end{aligned}
$$

B.4.2 Derivatives

Note that $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_d)^\top$ and $\mathbf{\Xi} = \mathrm{diag}[\xi_1, \ldots, \xi_d]^\top$, as defined previously in Section 3.5.

From Appendix B.4.1,

$$
\omega_{\mathbf{z}\mathbf{x}} = \alpha \prod_{k=1}^{d} (\xi_k x_k^2 + 1)^{-\frac{1}{2}} \exp\left(-\frac{(x_k \nu_k - z_k)^2}{2(\xi_k x_k^2 + 1)} \right)
$$

where $\mathbf{z} = (z_1, \ldots, z_d)^\top$ and $\mathbf{x} = (x_1, \ldots, x_d)^\top$. The partial derivative of $\omega_{\mathbf{zx}}$ with respect to $\boldsymbol{\nu}$, $\boldsymbol{\Xi}$, α, and β can be derived as follows:

$$\begin{aligned}\frac{\partial \omega_{\mathbf{zx}}}{\partial \nu_i} &= \alpha \prod_{k=1}^{d} (\xi_k x_k^2 + 1)^{-\frac{1}{2}} \exp\left(-\frac{(x_k\nu_k - z_k)^2}{2(\xi_k x_k^2 + 1)}\right) \times \left(-\frac{(x_i\nu_i - z_i)^2}{2(\xi_i x_i^2 + 1)}\right)' \\ &= \alpha \prod_{k=1}^{d} (\xi_k x_k^2 + 1)^{-\frac{1}{2}} \exp\left(-\frac{(x_k\nu_k - z_k)^2}{2(\xi_k x_k^2 + 1)}\right) \times \left(\frac{-\nu_i x_i^2 + z_i x_i}{\xi_i x_i^2 + 1}\right),\end{aligned}$$

$$\begin{aligned}\frac{\partial \omega_{\mathbf{zx}}}{\partial \xi_i} &= \alpha \prod_{k\neq i} (\xi_k x_k^2 + 1)^{-\frac{1}{2}} \exp\left(-\frac{(x_k\nu_k - z_k)^2}{2(\xi_k x_k^2 + 1)}\right) \times \left((\xi_i x_i^2 + 1)^{-\frac{1}{2}} \exp\left(-\frac{(x_i\nu_i - z_i)^2}{2(\xi_i x_i^2 + 1)}\right)\right)' \\ &= \alpha \prod_{k\neq i} (\xi_k x_k^2 + 1)^{-\frac{1}{2}} \exp\left(-\frac{(x_k\nu_k - z_k)^2}{2(\xi_k x_k^2 + 1)}\right) \times \left\{\left((\xi_i x_i^2 + 1)^{-\frac{1}{2}}\right)' \exp\left(-\frac{(x_i\nu_i - z_i)^2}{2(\xi_i x_i^2 + 1)}\right)\right. \\ &\qquad \left. + (\xi_i x_i^2 + 1)^{-\frac{1}{2}} \left(\exp\left(-\frac{(x_i\nu_i - z_i)^2}{2(\xi_i x_i^2 + 1)}\right)\right)'\right\} \\ &= \alpha \prod_{k=1}^{d} (\xi_k x_k^2 + 1)^{-\frac{1}{2}} \exp\left(-\frac{(x_k\nu_k - z_k)^2}{2(\xi_k x_k^2 + 1)}\right) \times \left(-\frac{x_i^2}{2(\xi_i x_i^2 + 1)} + \frac{x_i^2 (x_i\nu_i - z_i)^2}{2(\xi_i x_i^2 + 1)^2}\right),\end{aligned}$$

$$\begin{aligned}\frac{\partial \omega_{\mathbf{zx}}}{\partial \alpha} &= \prod_{k=1}^{d} (\xi_k x_k^2 + 1)^{-\frac{1}{2}} \exp\left(-\frac{(x_k\nu_k - z_k)^2}{2(\xi_k x_k^2 + 1)}\right), \\ \frac{\partial \omega_{\mathbf{zx}}}{\partial \beta} &= 0\ .\end{aligned}$$

From Appendix B.4.1,

$$\begin{aligned}\psi_{\mathbf{zz}'} = \sum_{i=1}^{B} \sum_{\mathbf{x},\mathbf{x}'\in\mathcal{D}_i} (\beta + \alpha^2)\ c^i_{\mathbf{xx}'} \prod_{k=1}^{d} \Bigg\{ &(\xi_k(x_k^2 + x_k'^2) + 1)^{-\frac{1}{2}} \\ &\exp\left(-\frac{\xi_k(z_k' x_k - z_k x_k')^2 + (x_k\nu_k - z_k)^2 + (x_k'\nu_k - z_k')^2}{2(\xi_k(x_k^2 + x_k'^2) + 1)}\right)\Bigg\}\end{aligned}$$

where $\mathbf{z}' \triangleq (z_1', \ldots, z_d')^\top$ and $\mathbf{x}' \triangleq (x_1', \ldots, x_d')^\top$. The partial derivative of $\psi_{\mathbf{xx}'}$ with

respect to $\boldsymbol{\nu}$, $\boldsymbol{\Xi}$, α, and β can be derived as follows:

$$\begin{aligned}
\frac{\partial \psi_{\mathbf{zz'}}}{\partial \nu_i} &= \sum_{j=1}^{B} \sum_{\mathbf{x},\mathbf{x'} \in \mathcal{D}_j} \Bigg[\Big(\beta + \alpha^2\Big) c^j_{\mathbf{xx'}} \prod_{k=1}^{d} \Bigg\{ (\xi_k(x_k^2 + x_k'^2) + 1)^{-\frac{1}{2}} \\
&\quad \exp\left(-\frac{\xi_k(z'_k x_k - z_k x'_k)^2 + (x_k\nu_k - z_k)^2 + (x'_k\nu_k - z'_k)^2}{2(\xi_k(x_k^2 + x_k'^2) + 1)} \right) \Bigg\} \\
&\quad \times \left(-\frac{\xi_i(z'_i x_i - z_i x'_i)^2 + (x_i\nu_i - z_i)^2 + (x'_i\nu_i - z'_i)^2}{2(\xi_i(x_i^2 + x_i'^2) + 1)} \right)' \Bigg] \\
&= \sum_{j=1}^{B} \sum_{\mathbf{x},\mathbf{x'} \in \mathcal{D}_j} \Bigg[\Big(\beta + \alpha^2\Big) c^j_{\mathbf{xx'}} \prod_{k=1}^{d} \Bigg\{ (\xi_k(x_k^2 + x_k'^2) + 1)^{-\frac{1}{2}} \\
&\quad \exp\left(-\frac{\xi_k(z'_k x_k - z_k x'_k)^2 + (x_k\nu_k - z_k)^2 + (x'_k\nu_k - z'_k)^2}{2(\xi_k(x_k^2 + x_k'^2) + 1)} \right) \Bigg\} \\
&\quad \times \left(-\frac{\nu_i(x_i^2 + x_i'^2) - (z_i x_i + z'_i x'_i)}{\xi_i(x_i^2 + x_i'^2) + 1} \right) \Bigg],
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \psi_{\mathbf{zz'}}}{\partial \xi_i} &= \sum_{j=1}^{B} \sum_{\mathbf{x},\mathbf{x'} \in \mathcal{D}_j} \Bigg[\Big(\beta + \alpha^2\Big) c^j_{\mathbf{xx'}} \prod_{k \neq i} \Bigg\{ (\xi_k(x_k^2 + x_k'^2) + 1)^{-\frac{1}{2}} \\
&\quad \exp\left(-\frac{\xi_k(z'_k x_k - z_k x'_k)^2 + (x_k\nu_k - z_k)^2 + (x'_k\nu_k - z'_k)^2}{2(\xi_k(x_k^2 + x_k'^2) + 1)} \right) \Bigg\} \\
&\quad \times \left((\xi_i(x_i^2 + x_i'^2) + 1)^{-\frac{1}{2}} \times \exp\left(-\frac{\xi_i(z'_i x_i - z_i x'_i)^2 + (x_i\nu_i - z_i)^2 + (x'_i\nu_i - z'_i)^2}{2(\xi_i(x_i^2 + x_i'^2) + 1)} \right) \right)' \Bigg] \\
&= \sum_{j=1}^{B} \sum_{\mathbf{x},\mathbf{x'} \in \mathcal{D}_j} \Bigg[\Big(\beta + \alpha^2\Big) c^j_{\mathbf{xx'}} \prod_{k=1}^{d} \Bigg\{ (\xi_k(x_k^2 + x_k'^2) + 1)^{-\frac{1}{2}} \\
&\quad \exp\left(-\frac{\xi_k(z'_k x_k - z_k x'_k)^2 + (x_k\nu_k - z_k)^2 + (x'_k\nu_k - z'_k)^2}{2(\xi_k(x_k^2 + x_k'^2) + 1)} \right) \Bigg\} \\
&\quad \times \left(-\frac{x_i^2 + x_i'^2}{2\Big(\xi_i(x_i^2 + x_i'^2) + 1\Big)} + \frac{\Big(z_i x_i + z'_i x'_i - \nu_i(x_i^2 + x_i'^2)\Big)^2}{2\Big(\xi_i(x_i^2 + x_i'^2) + 1\Big)^2} \right) \Bigg],
\end{aligned}$$

$$\frac{\partial \psi_{\mathbf{zz'}}}{\partial \alpha} = \sum_{i=1}^{B} \sum_{\mathbf{x},\mathbf{x'} \in \mathcal{D}_i} 2\alpha \; c^i_{\mathbf{xx'}} \prod_{k=1}^{d} \Bigg\{ (\xi_k(x_k^2 + x_k'^2) + 1)^{-\frac{1}{2}} \\ \exp\left(-\frac{\xi_k(z_k' x_k - z_k x_k')^2 + (x_k \nu_k - z_k)^2 + (x_k' \nu_k - z_k')^2}{2(\xi_k(x_k^2 + x_k'^2) + 1)} \right) \Bigg\},$$

$$\frac{\partial \psi_{\mathbf{zz'}}}{\partial \beta} = \sum_{i=1}^{B} \sum_{\mathbf{x},\mathbf{x'} \in \mathcal{D}_i} c^i_{\mathbf{xx'}} \prod_{k=1}^{d} \Bigg\{ (\xi_k(x_k^2 + x_k'^2) + 1)^{-\frac{1}{2}} \\ \exp\left(-\frac{\xi_k(z_k' x_k - z_k x_k')^2 + (x_k \nu_k - z_k)^2 + (x_k' \nu_k - z_k')^2}{2(\xi_k(x_k^2 + x_k'^2) + 1)} \right) \Bigg\}.$$

From Appendix B.4.1,

$$\gamma_{\mathbf{xx'}} = \left(\beta + \alpha^2\right) \prod_{k=1}^{d} (\xi_k(x_k - x_k')^2 + 1)^{-\frac{1}{2}} \exp\left(-\frac{\nu_k^2 (x_k - x_k')^2}{2(\xi_k(x_k - x_k')^2 + 1)} \right).$$

The partial derivative of $\gamma_{\mathbf{xx'}}$ with respect to $\boldsymbol{\nu}$, $\boldsymbol{\Xi}$, α, and β can be derived as follows:

$$\begin{aligned} \frac{\partial \gamma_{\mathbf{xx'}}}{\partial \nu_i} &= \left(\beta + \alpha^2\right) \prod_{k=1}^{d} (\xi_k(x_k - x_k')^2 + 1)^{-\frac{1}{2}} \exp\left(-\frac{\nu_k^2 (x_k - x_k')^2}{2(\xi_k(x_k - x_k')^2 + 1)} \right) \\ &\quad \times \left(-\frac{\nu_i^2 (x_i - x_i')^2}{2\left(\xi_i(x_i - x_i')^2 + 1\right)} \right)' \\ &= \left(\beta + \alpha^2\right) \prod_{k=1}^{d} (\xi_k(x_k - x_k')^2 + 1)^{-\frac{1}{2}} \exp\left(-\frac{\nu_k^2 (x_k - x_k')^2}{2(\xi_k(x_k - x_k')^2 + 1)} \right) \\ &\quad \times \left(-\frac{\nu_i (x_i - x_i')^2}{\xi_i(x_i - x_i')^2 + 1} \right), \end{aligned}$$

$$
\begin{aligned}
\frac{\partial \gamma_{\mathbf{x}\mathbf{x}'}}{\partial \xi_i} &= \left(\beta + \alpha^2\right) \prod_{k \neq i} \left(\xi_k (x_k - x'_k)^2 + 1\right)^{-\frac{1}{2}} \exp\left(-\frac{\nu_k^2 (x_k - x'_k)^2}{2(\xi_k (x_k - x'_k)^2 + 1)}\right) \\
&\quad \times \Bigg(\left(\left(\xi_i (x_i - x'_i)^2 + 1\right)^{-\frac{1}{2}}\right)' \exp\left(-\frac{\nu_i^2 (x_i - x'_i)^2}{2(\xi_i (x_i - x'_i)^2 + 1)}\right) \\
&\quad + \left(\xi_i (x_i - x'_i)^2 + 1\right)^{-\frac{1}{2}} \left(\exp\left(-\frac{\nu_i^2 (x_i - x'_i)^2}{2(\xi_i (x_i - x'_i)^2 + 1)}\right)\right)' \Bigg) \\
&= \left(\beta + \alpha^2\right) \prod_{k=1}^{d} \left(\xi_k (x_k - x'_k)^2 + 1\right)^{-\frac{1}{2}} \exp\left(-\frac{\nu_k^2 (x_k - x'_k)^2}{2(\xi_k (x_k - x'_k)^2 + 1)}\right) \\
&\quad \times \left(-\frac{(x_i - x'_i)^2}{2\left(\xi_i (x_i - x'_i)^2 + 1\right)} + \frac{\nu_i^2 (x_i - x'_i)^4}{2\left(\xi_i (x_i - x'_i)^2 + 1\right)^2}\right) \\
&= \left(\beta + \alpha^2\right) \prod_{k=1}^{d} \left(\xi_k (x_k - x'_k)^2 + 1\right)^{-\frac{1}{2}} \exp\left(-\frac{\nu_k^2 (x_k - x'_k)^2}{2(\xi_k (x_k - x'_k)^2 + 1)}\right) \\
&\quad \times \frac{(\nu_i^2 - \xi_i)(x_i - x'_i)^4 - (x_i - x'_i)^2}{2\left(\xi_i (x_i - x'_i)^2 + 1\right)^2} \,,
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \gamma_{\mathbf{x}\mathbf{x}'}}{\partial \alpha} &= 2\alpha \prod_{k=1}^{d} \left(\xi_k (x_k - x'_k)^2 + 1\right)^{-\frac{1}{2}} \exp\left(-\frac{\nu_k^2 (x_k - x'_k)^2}{2(\xi_k (x_k - x'_k)^2 + 1)}\right), \\
\frac{\partial \gamma_{\mathbf{x}\mathbf{x}'}}{\partial \beta} &= \prod_{k=1}^{d} \left(\xi_k (x_k - x'_k)^2 + 1\right)^{-\frac{1}{2}} \exp\left(-\frac{\nu_k^2 (x_k - x'_k)^2}{2(\xi_k (x_k - x'_k)^2 + 1)}\right).
\end{aligned}
$$

B.5 VBSGP Predictive Mean and Variance

B.5.1 VBPITC, VBFIC, VBFITC, and VBDTC

VBPITC, VBFIC, VBFITC, and VBDTC share the same approximated test conditional $q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}_B}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f) \triangleq p(f_{\mathbf{x}^*}|\mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)$ but differ in $q^+(\mathbf{s}_{\mathcal{I}})$, $q^+(\mathbf{\Lambda})$, and $q^+(\sigma_f)$ obtained from their stochastic gradient ascent updates. As a result,

$$
q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}}) = \int p(f_{\mathbf{x}^*}|\mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)\; q^+(\mathbf{s}_{\mathcal{I}})\; q^+(\mathbf{\Lambda})\; q^+(\sigma_f)\; \mathrm{d}\mathbf{s}_{\mathcal{I}}\; \mathrm{d}\mathbf{\Lambda}\; \mathrm{d}\sigma_f
$$

where

$$p(f_{\mathbf{x}^*}|\mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f) = \mathcal{N}(\mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{s}_{\mathcal{I}}, k_{\mathbf{x}^*\mathbf{x}^*} - \mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{K}_{\mathcal{I}\mathbf{x}^*}),\ q^+(\mathbf{s}_{\mathcal{I}}) = \mathcal{N}(\mathbf{m}^+, \mathbf{S}^+)\ ,$$
$$q^+(\mathbf{\Lambda}) = \prod_{i=1}^{d} \mathcal{N}(\lambda_i|\nu_i^+, \xi_i^+),\ q^+(\sigma_f) = \mathcal{N}(\alpha^+, \beta^+)\ .$$

Then,

$$\begin{aligned} q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}}, \mathbf{\Lambda}, \sigma_f) &= \int p(f_{\mathbf{x}^*}|\mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)\ q^+(\mathbf{s}_{\mathcal{I}})\ \mathrm{d}\mathbf{s}_{\mathcal{I}} \\ &= \mathcal{N}(\mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m}^+, k_{\mathbf{x}^*\mathbf{x}^*} - \mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{K}_{\mathcal{I}\mathbf{x}^*} + \mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{S}^+\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{K}_{\mathcal{I}\mathbf{x}^*})\ . \end{aligned}$$

Finally,

$$\begin{aligned} \mu_{\mathbf{x}^*|\mathcal{D}} &\triangleq \mathbb{E}_{q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}})}[f_{\mathbf{x}^*}] = \mathbb{E}_{q^+(\mathbf{\Lambda},\sigma_f)}[\mathbb{E}_{q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}},\mathbf{\Lambda},\sigma_f)}[f_{\mathbf{x}^*}]] \\ &= \mathbb{E}_{q^+(\mathbf{\Lambda},\sigma_f)}[\mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m}^+] = \mathbb{E}_{q^+(\mathbf{\Lambda},\sigma_f)}[\mathbf{K}_{\mathbf{x}^*\mathcal{I}}]\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m}^+\ . \end{aligned}$$

$$\begin{aligned} &\sigma_{\mathbf{x}^*|\mathcal{D}}^2 \triangleq \mathbb{V}_{q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}})}[f_{\mathbf{x}^*}] \\ &\quad = \mathbb{E}_{q^+(\mathbf{\Lambda},\sigma_f)}[\mathbb{V}_{q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}},\mathbf{\Lambda},\sigma_f)}[f_{\mathbf{x}^*}]] + \mathbb{V}_{q^+(\mathbf{\Lambda},\sigma_f)}[\mathbb{E}_{q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}},\mathbf{\Lambda},\sigma_f)}[f_{\mathbf{x}^*}]] \\ &\quad = \mathbb{E}_{q^+(\mathbf{\Lambda},\sigma_f)}[k_{\mathbf{x}^*\mathbf{x}^*} - \mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{K}_{\mathcal{I}\mathbf{x}^*} + \mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{S}^+\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{K}_{\mathcal{I}\mathbf{x}^*}] + \mathbb{V}_{q^+(\mathbf{\Lambda},\sigma_f)}[\mathbf{m}^{+\top}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{K}_{\mathcal{I}\mathbf{x}^*}] \end{aligned}$$

where

$$\begin{aligned} &\mathbb{V}_{q^+(\mathbf{\Lambda},\sigma_f)}[\mathbf{m}^{+\top}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{K}_{\mathcal{I}\mathbf{x}^*}] = \mathbf{m}^{+\top}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbb{V}_{q^+(\mathbf{\Lambda},\sigma_f)}[\mathbf{K}_{\mathcal{I}\mathbf{x}^*}]\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m}^+ \\ &= \mathbf{m}^{+\top}\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\Big(\mathbb{E}_{q^+(\mathbf{\Lambda},\sigma_f)}[\mathbf{K}_{\mathcal{I}\mathbf{x}^*}\mathbf{K}_{\mathbf{x}^*\mathcal{I}}] - \mathbb{E}_{q^+(\mathbf{\Lambda},\sigma_f)}[\mathbf{K}_{\mathcal{I}\mathbf{x}^*}]\mathbb{E}_{q^+(\mathbf{\Lambda},\sigma_f)}[\mathbf{K}_{\mathbf{x}^*\mathcal{I}}]\Big)\mathbf{\Sigma}_{\mathcal{I}\mathcal{I}}^{-1}\mathbf{m}^+. \end{aligned}$$

Note that the closed-form expressions of all the above expectation terms with respect to $q^+(\mathbf{\Lambda}, \sigma_f) \triangleq q^+(\mathbf{\Lambda})q^+(\sigma_f)$ can be derived in a similar manner as that of $\mathbf{\Psi}_{\mathcal{I}\mathcal{I}} \triangleq \mathbb{E}_{q(\mathbf{\Lambda},\sigma_f)}[\mathbf{K}_{\mathcal{I}\mathcal{D}}\mathbf{C}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{K}_{\mathcal{D}\mathcal{I}}]$, $\mathbf{\Omega}_{\mathcal{I}\mathcal{D}} \triangleq \mathbb{E}_{q(\mathbf{\Lambda},\sigma_f)}[\mathbf{K}_{\mathcal{I}\mathcal{D}}]$, and $\mathbf{\Upsilon}_{\mathcal{D}\mathcal{D}} \triangleq \mathbb{E}_{q(\mathbf{\Lambda},\sigma_f)}[\mathbf{K}_{\mathcal{D}\mathcal{D}}]$. Hence, $\mu_{\mathbf{x}^*|\mathcal{D}}$ and $\sigma_{\mathbf{x}^*|\mathcal{D}}^2$ can be derived in closed form.

B.5.2 VBPIC

VBPIC uses the exact test conditional $q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}_B}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f) \triangleq p(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}_B}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)$. To derive $p(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}_B}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)$, we use the fundamental definition of GP to give the following expression for the Gaussian joint distribution $p(f_{\mathbf{x}^*}, \mathbf{s}_{\mathcal{I}}, \mathbf{y}_{\mathcal{D}_B}|\mathbf{\Lambda}, \sigma_f)$:

$$\mathcal{N}\left(\mathbf{0}, \begin{pmatrix} k_{\mathbf{x}^*\mathbf{x}^*} & \mathbf{K}_{\mathbf{x}^*\mathcal{I}} & \mathbf{K}_{\mathbf{x}^*\mathcal{D}_B} \\ \mathbf{K}_{\mathcal{I}\mathbf{x}^*} & \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}} & \mathbf{K}_{\mathcal{I}\mathcal{D}_B} \\ \mathbf{K}_{\mathcal{D}_B\mathbf{x}^*} & \mathbf{K}_{\mathcal{D}_B\mathcal{I}} & \mathbf{K}_{\mathcal{D}_B\mathcal{D}_B}+\mathbf{C}_{\mathcal{D}_B\mathcal{D}_B} \end{pmatrix}\right).$$

Then, $p(f_{\mathbf{x}^*}|\mathbf{s}_{\mathcal{I}}, \mathbf{y}_{\mathcal{D}_B}, \mathbf{\Lambda}, \sigma_f) = \mathcal{N}(\mathbb{E}_{p(f_{\mathbf{x}^*}|\mathbf{s}_{\mathcal{I}}, \mathbf{y}_{\mathcal{D}_B}, \mathbf{\Lambda}, \sigma_f)}[f_{\mathbf{x}^*}], \mathbb{V}_{p(f_{\mathbf{x}^*}|\mathbf{s}_{\mathcal{I}}, \mathbf{y}_{\mathcal{D}_B}, \mathbf{\Lambda}, \sigma_f)}[f_{\mathbf{x}^*}])$ where

$$\mathbb{E}_{p(f_{\mathbf{x}^*}|\mathbf{s}_{\mathcal{I}}, \mathbf{y}_{\mathcal{D}_B}, \mathbf{\Lambda}, \sigma_f)}[f_{\mathbf{x}^*}] = \begin{pmatrix} \mathbf{K}_{\mathbf{x}^*\mathcal{I}} & \mathbf{K}_{\mathbf{x}^*\mathcal{D}_B} \end{pmatrix} \begin{pmatrix} \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}} & \mathbf{K}_{\mathcal{I}\mathcal{D}_B} \\ \mathbf{K}_{\mathcal{D}_B\mathcal{I}} & \mathbf{K}_{\mathcal{D}_B\mathcal{D}_B} + \mathbf{C}_{\mathcal{D}_B\mathcal{D}_B} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{s}_{\mathcal{I}} \\ \mathbf{y}_{\mathcal{D}_B} \end{pmatrix},$$

$$\mathbb{V}_{p(f_{\mathbf{x}^*}|\mathbf{s}_{\mathcal{I}}, \mathbf{y}_{\mathcal{D}_B}, \mathbf{\Lambda}, \sigma_f)}[f_{\mathbf{x}^*}] = k_{\mathbf{x}^*\mathbf{x}^*} - \begin{pmatrix} \mathbf{K}_{\mathbf{x}^*\mathcal{I}} & \mathbf{K}_{\mathbf{x}^*\mathcal{D}_B} \end{pmatrix} \begin{pmatrix} \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}} & \mathbf{K}_{\mathcal{I}\mathcal{D}_B} \\ \mathbf{K}_{\mathcal{D}_B\mathcal{I}} & \mathbf{K}_{\mathcal{D}_B\mathcal{D}_B}+\mathbf{C}_{\mathcal{D}_B\mathcal{D}_B} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{K}_{\mathcal{I}\mathbf{x}^*} \\ \mathbf{K}_{\mathcal{D}_B\mathbf{x}^*} \end{pmatrix}.$$

To simplify the above expressions, let

$$\mathbf{J} \triangleq \begin{pmatrix} \mathbf{\Sigma}_{\mathcal{I}\mathcal{I}} & \mathbf{K}_{\mathcal{I}\mathcal{D}_B} \\ \mathbf{K}_{\mathcal{D}_B\mathcal{I}} & \mathbf{K}_{\mathcal{D}_B\mathcal{D}_B} + \mathbf{C}_{\mathcal{D}_B\mathcal{D}_B} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{J}_{\mathcal{I}\mathcal{I}} & \mathbf{J}_{\mathcal{I}\mathcal{D}_B} \\ \mathbf{J}_{\mathcal{D}_B\mathcal{I}} & \mathbf{J}_{\mathcal{D}_B\mathcal{D}_B} \end{pmatrix}$$

where $\mathbf{J}_{\mathcal{I}\mathcal{I}}$, $\mathbf{J}_{\mathcal{I}\mathcal{D}_B}$, $\mathbf{J}_{\mathcal{D}_B\mathcal{I}}$, and $\mathbf{J}_{\mathcal{D}_B\mathcal{D}_B}$ can be derived by applying the matrix inversion lemma for partitioned matrices directly. Then,

$$\begin{aligned} \mathbb{E}_{p(f_{\mathbf{x}^*}|\mathbf{s}_{\mathcal{I}}, \mathbf{y}_{\mathcal{D}_B}, \mathbf{\Lambda}, \sigma_f)}[f_{\mathbf{x}^*}] &= \begin{pmatrix} \mathbf{K}_{\mathbf{x}^*\mathcal{I}} & \mathbf{K}_{\mathbf{x}^*\mathcal{D}_B} \end{pmatrix} \begin{pmatrix} \mathbf{J}_{\mathcal{I}\mathcal{I}} & \mathbf{J}_{\mathcal{I}\mathcal{D}_B} \\ \mathbf{J}_{\mathcal{D}_B\mathcal{I}} & \mathbf{J}_{\mathcal{D}_B\mathcal{D}_B} \end{pmatrix} \begin{pmatrix} \mathbf{s}_{\mathcal{I}} \\ \mathbf{y}_{\mathcal{D}_B} \end{pmatrix} \\ &= \left(\mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{J}_{\mathcal{I}\mathcal{I}} + \mathbf{K}_{\mathbf{x}^*\mathcal{D}_B}\mathbf{J}_{\mathcal{D}_B\mathcal{I}}\right)\mathbf{s}_{\mathcal{I}} + \left(\mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{J}_{\mathcal{I}\mathcal{D}_B} + \mathbf{K}_{\mathbf{x}^*\mathcal{D}_B}\mathbf{J}_{\mathcal{D}_B\mathcal{D}_B}\right)\mathbf{y}_{\mathcal{D}_B}\ , \\ \mathbb{V}_{p(f_{\mathbf{x}^*}|\mathbf{s}_{\mathcal{I}}, \mathbf{y}_{\mathcal{D}_B}, \mathbf{\Lambda}, \sigma_f)}[f_{\mathbf{x}^*}] &= k_{\mathbf{x}^*\mathbf{x}^*} - \mathbf{K}_{\mathbf{x}^*(\mathcal{I}\cup\mathcal{D}_B)}\mathbf{J}\mathbf{K}_{(\mathcal{I}\cup\mathcal{D}_B)\mathbf{x}^*}\ . \end{aligned}$$

Now,

$$q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}}) = \int p(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}_B}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)\ q^+(\mathbf{s}_{\mathcal{I}})\ q^+(\mathbf{\Lambda})\ q^+(\sigma_f) \mathrm{d}\mathbf{s}_{\mathcal{I}} \mathrm{d}\mathbf{\Lambda} \mathrm{d}\sigma_f$$

where

$$\begin{aligned}
&p(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}_B}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f) = \mathcal{N}(f_{\mathbf{x}^*}|\mathbb{E}_{p(f_{\mathbf{x}^*}|\mathbf{s}_{\mathcal{I}}, \mathbf{y}_{\mathcal{D}_B}, \mathbf{\Lambda}, \sigma_f)}[f_{\mathbf{x}^*}], \mathbb{V}_{p(f_{\mathbf{x}^*}|\mathbf{s}_{\mathcal{I}}, \mathbf{y}_{\mathcal{D}_B}, \mathbf{\Lambda}, \sigma_f)}[f_{\mathbf{x}^*}])\ , \\
&q^+(\mathbf{s}_{\mathcal{I}}) = \mathcal{N}(\mathbf{m}^+, \mathbf{S}^+)\ ,\ q^+(\mathbf{\Lambda}) = \prod_{i=1}^{d} \mathcal{N}(\lambda_i|\nu_i^+, \xi_i^+)\ ,\ q^+(\sigma_f) = \mathcal{N}(\alpha^+, \beta^+)\ .
\end{aligned}$$

Then,

$$\begin{aligned}
&q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}}, \mathbf{\Lambda}, \sigma_f) = \int p(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}_B}, \mathbf{s}_{\mathcal{I}}, \mathbf{\Lambda}, \sigma_f)\ q^+(\mathbf{s}_{\mathcal{I}})\ \mathrm{d}\mathbf{s}_{\mathcal{I}} \\
&= \mathcal{N}((\mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{J}_{\mathcal{I}\mathcal{I}} + \mathbf{K}_{\mathbf{x}^*\mathcal{D}_B}\mathbf{J}_{\mathcal{D}_B\mathcal{I}})\,\mathbf{m}^+ + (\mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{J}_{\mathcal{I}\mathcal{D}_B} + \mathbf{K}_{\mathbf{x}^*\mathcal{D}_B}\mathbf{J}_{\mathcal{D}_B\mathcal{D}_B})\,\mathbf{y}_{\mathcal{D}_B}\ , \\
&k_{\mathbf{x}^*\mathbf{x}^*} - \mathbf{K}_{\mathbf{x}^*(\mathcal{I}\cup\mathcal{D}_B)}\mathbf{J}\mathbf{K}_{(\mathcal{I}\cup\mathcal{D}_B)\mathbf{x}^*} + (\mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{J}_{\mathcal{I}\mathcal{I}} + \mathbf{K}_{\mathbf{x}^*\mathcal{D}_B}\mathbf{J}_{\mathcal{D}_B\mathcal{I}})\,\mathbf{S}^+\,(\mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{J}_{\mathcal{I}\mathcal{I}} + \mathbf{K}_{\mathbf{x}^*\mathcal{D}_B}\mathbf{J}_{\mathcal{D}_B\mathcal{I}})^\top)\ .
\end{aligned}$$

Finally,

$$\begin{aligned}
&\mu_{\mathbf{x}^*|\mathcal{D}} \triangleq \mathbb{E}_{q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}})}[f_{\mathbf{x}^*}] \\
&= \mathbb{E}_{q^+(\mathbf{\Lambda},\sigma_f)}[\mathbb{E}_{q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}},\mathbf{\Lambda},\sigma_f)}[f_{\mathbf{x}^*}]] \\
&= \mathbb{E}_{q^+(\mathbf{\Lambda},\sigma_f)}[(\mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{J}_{\mathcal{I}\mathcal{I}} + \mathbf{K}_{\mathbf{x}^*\mathcal{D}_B}\mathbf{J}_{\mathcal{D}_B\mathcal{I}})\,\mathbf{m}^+ + (\mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{J}_{\mathcal{I}\mathcal{D}_B} + \mathbf{K}_{\mathbf{x}^*\mathcal{D}_B}\mathbf{J}_{\mathcal{D}_B\mathcal{D}_B})\,\mathbf{y}_{\mathcal{D}_B}] \\
&= \mathbb{E}_{q^+(\mathbf{\Lambda},\sigma_f)}\left[\mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{J}_{\mathcal{I}\mathcal{I}} + \mathbf{K}_{\mathbf{x}^*\mathcal{D}_B}\mathbf{J}_{\mathcal{D}_B\mathcal{I}}\right]\mathbf{m}^+ + \ \mathbb{E}_{q^+(\mathbf{\Lambda},\sigma_f)}\left[\mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{J}_{\mathcal{I}\mathcal{D}_B} + \mathbf{K}_{\mathbf{x}^*\mathcal{D}_B}\mathbf{J}_{\mathcal{D}_B\mathcal{D}_B}\right]\mathbf{y}_{\mathcal{D}_B}\ .
\end{aligned}$$

$$\begin{aligned}
&\sigma^2_{\mathbf{x}^*|\mathcal{D}} \triangleq \mathbb{V}_{q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}})}[f_{\mathbf{x}^*}] \\
&= \mathbb{E}_{q^+(\mathbf{\Lambda},\sigma_f)}[\mathbb{V}_{q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}},\mathbf{\Lambda},\sigma_f)}[f_{\mathbf{x}^*}]] + \mathbb{V}_{q^+(\mathbf{\Lambda},\sigma_f)}[\mathbb{E}_{q(f_{\mathbf{x}^*}|\mathbf{y}_{\mathcal{D}},\mathbf{\Lambda},\sigma_f)}[f_{\mathbf{x}^*}]] \\
&= \mathbb{E}_{q^+(\mathbf{\Lambda},\sigma_f)}\Bigg[k_{\mathbf{x}^*\mathbf{x}^*} - \mathbf{K}_{\mathbf{x}^*(\mathcal{I}\cup\mathcal{D}_B)}\mathbf{J}\mathbf{K}_{(\mathcal{I}\cup\mathcal{D}_B)\mathbf{x}^*} \\
&\qquad\qquad + (\mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{J}_{\mathcal{I}\mathcal{I}} + \mathbf{K}_{\mathbf{x}^*\mathcal{D}_B}\mathbf{J}_{\mathcal{D}_B\mathcal{I}})\,\mathbf{S}^+\,(\mathbf{K}_{\mathbf{x}^*\mathcal{I}}\mathbf{J}_{\mathcal{I}\mathcal{I}} + \mathbf{K}_{\mathbf{x}^*\mathcal{D}_B}\mathbf{J}_{\mathcal{D}_B\mathcal{I}})^\top\Bigg] \\
&\quad + \mathbb{V}_{q^+(\mathbf{\Lambda},\sigma_f)}\left[(\mathbf{m}^{+\top}\ \mathbf{y}_{\mathcal{D}_B}^\top)\mathbf{J}\mathbf{K}_{(\mathcal{I}\cup\mathcal{D}_B)\mathbf{x}^*}\right]
\end{aligned}$$

where

$$\mathbb{V}_{q^+(\boldsymbol{\Lambda},\sigma_f)}\left[(\mathbf{m}^{+\top}\ \mathbf{y}_{\mathcal{D}_B}^\top)\mathbf{J}\mathbf{K}_{(\mathcal{I}\cup\mathcal{D}_B)\mathbf{x}^*}\right] = (\mathbf{m}^{+\top}\ \mathbf{y}_{\mathcal{D}_B}^\top)\mathbb{V}_{q^+(\boldsymbol{\Lambda},\sigma_f)}\left[\mathbf{J}\mathbf{K}_{(\mathcal{I}\cup\mathcal{D}_B)\mathbf{x}^*}\right]\begin{pmatrix}\mathbf{m}^+\\ \mathbf{y}_{\mathcal{D}_B}\end{pmatrix}$$
$$= (\mathbf{m}^{+\top}\ \mathbf{y}_{\mathcal{D}_B}^\top)\Big(\mathbb{E}_{q^+(\boldsymbol{\Lambda},\sigma_f)}\left[\mathbf{J}\mathbf{K}_{(\mathcal{I}\cup\mathcal{D}_B)\mathbf{x}^*}\mathbf{K}_{\mathbf{x}^*(\mathcal{I}\cup\mathcal{D}_B)}\mathbf{J}\right]$$
$$-\mathbb{E}_{q^+(\boldsymbol{\Lambda},\sigma_f)}[\mathbf{J}\mathbf{K}_{(\mathcal{I}\cup\mathcal{D}_B)\mathbf{x}^*}]\mathbb{E}_{q^+(\boldsymbol{\Lambda},\sigma_f)}[\mathbf{K}_{\mathbf{x}^*(\mathcal{I}\cup\mathcal{D}_B)}\mathbf{J}]\Big)\begin{pmatrix}\mathbf{m}^+\\ \mathbf{y}_{\mathcal{D}_B}\end{pmatrix}\ .$$

Unfortunately, the closed-form expressions of all the above expectation terms with respect to $q^+(\boldsymbol{\Lambda},\sigma_f) \triangleq q^+(\boldsymbol{\Lambda})q^+(\sigma_f)$ cannot be obtained because it involves integrating, over $\boldsymbol{\Lambda}$, terms containing $\mathbf{J}$ that depends on $\boldsymbol{\Lambda}$ but without an analytical form with respect to $\boldsymbol{\Lambda}$. So, we approximate them via sampling.

Appendix C

Appendix for Chapter 4

C.1 Proof of Proposition 1

The objective function in (4.3) can be re-written as

$$\int p(\boldsymbol{\mathcal{U}}) \log(1 - \sigma(T(\boldsymbol{\mathcal{U}}))) \, \mathrm{d}\boldsymbol{\mathcal{U}} + \int q_\Phi(\boldsymbol{\mathcal{U}}) \log \sigma(T(\boldsymbol{\mathcal{U}})) \, \mathrm{d}\boldsymbol{\mathcal{U}} \; .$$

The above integral is maximal in function T if and only if the integrand is maximal in $T(\boldsymbol{\mathcal{U}})$ for every $\boldsymbol{\mathcal{U}}$. Note that the maximum of $a \log(t) + b \log(1 - t)$ over $t \in [0, 1]$ is at $t = a/(a + b)$ for any $(a, b) \in \mathbb{R}^2 \backslash (0, 0)$. Using this result,

$$\sigma(T^*(\boldsymbol{\mathcal{U}})) = \frac{q_\Phi(\boldsymbol{\mathcal{U}})}{q_\Phi(\boldsymbol{\mathcal{U}}) + p(\boldsymbol{\mathcal{U}})}$$

or, equivalently,

$$T^*(\boldsymbol{\mathcal{U}}) = \log q_\Phi(\boldsymbol{\mathcal{U}}) - \log p(\boldsymbol{\mathcal{U}}) \; .$$

C.2 Proof of Proposition 2

If $(\{\Psi^*\}, \{\theta^*, \Phi^*\})$ is a Nash equilibrium, then according to Proposition 1 and under the assumption that T_{Ψ^*} is expressive enough, we know that **Player 1** is playing its optimal strategy Ψ^* such that

$$T_{\Psi^*}(\boldsymbol{\mathcal{U}}) = \log q_{\Phi^*}(\boldsymbol{\mathcal{U}}) - \log p(\boldsymbol{\mathcal{U}}) \ . \tag{C.1}$$

Substituting (C.1) into (4.5) reveals that **Player 2**'s strategy $\{\theta^*, \Phi^*\}$ maximizes its payoff which is a function of $\{\theta, \Phi\}$:

$$\begin{aligned}
\mathcal{F}(\theta, \Phi) &\triangleq \mathbb{E}_{q_\Phi(\boldsymbol{\mathcal{U}})}[\mathcal{L}(\theta, \mathbf{X}, \mathbf{y}, \boldsymbol{\mathcal{U}}) + \log p(\boldsymbol{\mathcal{U}}) - \log q_{\Phi^*}(\boldsymbol{\mathcal{U}})] \\
&= \mathbb{E}_{q_\Phi(\boldsymbol{\mathcal{U}})}[\mathcal{L}(\theta, \mathbf{X}, \mathbf{y}, \boldsymbol{\mathcal{U}}) + \log p(\boldsymbol{\mathcal{U}}) - \log q_{\Phi}(\boldsymbol{\mathcal{U}}) + \log q_{\Phi}(\boldsymbol{\mathcal{U}}) - \log q_{\Phi^*}(\boldsymbol{\mathcal{U}})] \\
&= \mathcal{EL}(\theta, \Phi) + \mathrm{KL}[q_\Phi(\boldsymbol{\mathcal{U}}) \| q_{\Phi^*}(\boldsymbol{\mathcal{U}})]
\end{aligned} \tag{C.2}$$

where $\mathcal{EL}(\theta, \Phi)$ is the ELBO in (4.2).

Now, suppose that $\{\theta^*, \Phi^*\}$ does not maximize the ELBO. Then, there exists some $\{\theta', \Phi'\}$ such that $\mathcal{EL}(\theta', \Phi') > \mathcal{EL}(\theta^*, \Phi^*)$. By substituting $\{\theta', \Phi'\}$ into (C.2),

$$\mathcal{F}(\theta', \Phi') = \mathcal{EL}(\theta', \Phi') + \mathrm{KL}[q_{\Phi'}(\boldsymbol{\mathcal{U}}) \| q_{\Phi^*}(\boldsymbol{\mathcal{U}})] > \mathcal{F}(\theta^*, \Phi^*) \ ,$$

which contradicts the fact that $\{\theta^*, \Phi^*\}$ maximizes (C.2). Hence, $\{\theta^*, \Phi^*\}$ maximizes the ELBO, which is equal to the log-marginal likelihood $\log p_{\theta^*}(\mathbf{y})$ with θ^* being the maximum likelihood assignment and $q_{\Phi^*}(\boldsymbol{\mathcal{U}})$ being equal to the true posterior belief $p(\boldsymbol{\mathcal{U}}|\mathbf{y})$.

C.3 Discussion on the Existence of Nash Equilibrium

Proposition 3. *Suppose that the parametric representations of T_Ψ and g_Φ are expressive enough to represent any function and the DGP model hyperparameters are fixed to be $\theta_\circ$. Then, the two-player pure-strategy game in* (4.6) *for the case of fixed $\theta_\circ$ has a Nash equilibrium. Furthermore, if $(\{\Psi^*\}, \{\theta_\circ, \Phi^*\})$ is a Nash equilibrium, then $\{\Phi^*\}$ is a global maximizer of the ELBO for the case of fixed $\theta_\circ$ such that $q_{\Phi^*}(\boldsymbol{\mathcal{U}})$ is equal to the true posterior belief $p_{\theta_\circ}(\boldsymbol{\mathcal{U}}|\mathbf{y})$.*

Proof. Since we assume the parametric representation of g_Φ to be expressive enough to represent any function, we can find some $\{\Phi_\circ\}$ such that $q_{\Phi_\circ}(\boldsymbol{\mathcal{U}})$ is equal to the true posterior belief $p_{\theta_\circ}(\boldsymbol{\mathcal{U}}|\mathbf{y})$. We now know that $\{\Phi_\circ\}$ maximizes the ELBO in (4.2) for the case of fixed DGP model hyperparameters $\theta_\circ$, which we denote by $\mathcal{EL}(\theta_\circ, \Phi_\circ)$.

Since we assume the parametric representation of T_Ψ to be expressive enough to represent any function, we can further obtain some $\{\Psi_\circ\}$ such that $T_{\Psi_\circ}(\boldsymbol{\mathcal{U}}) = \log q_{\Phi_\circ}(\boldsymbol{\mathcal{U}}) - \log p(\boldsymbol{\mathcal{U}})$. According to Proposition 1, $\{\Psi_\circ\}$ maximizes the payoff to **player 1**. Hence, **player 1** cannot improve its strategy to achieve a better payoff.

Given that **player 1** plays strategy $\{\Psi_\circ\}$ for the case of fixed $\theta_\circ$, the payoff to **player 2** playing strategy $\{\theta_\circ, \Phi\}$ is

$$
\begin{aligned}
\mathcal{F}(\theta_\circ, \Phi) &\triangleq \mathbb{E}_{q_\Phi(\boldsymbol{\mathcal{U}})}[\mathcal{L}(\theta_\circ, \mathbf{X}, \mathbf{y}, \boldsymbol{\mathcal{U}}) + \log p(\boldsymbol{\mathcal{U}}) - \log q_{\Phi_\circ}(\boldsymbol{\mathcal{U}})] \\
&= \mathbb{E}_{q_\Phi(\boldsymbol{\mathcal{U}})}[\mathcal{L}(\theta_\circ, \mathbf{X}, \mathbf{y}, \boldsymbol{\mathcal{U}}) + \log p(\boldsymbol{\mathcal{U}}) - \log q_\Phi(\boldsymbol{\mathcal{U}}) + \log q_\Phi(\boldsymbol{\mathcal{U}}) - \log q_{\Phi_\circ}(\boldsymbol{\mathcal{U}})] \\
&= \mathcal{EL}(\theta_\circ, \Phi) + \mathrm{KL}[q_\Phi(\boldsymbol{\mathcal{U}}) \| q_{\Phi_\circ}(\boldsymbol{\mathcal{U}})] \\
&= \log p_{\theta_\circ}(\mathbf{y}) - \mathrm{KL}[q_\Phi(\boldsymbol{\mathcal{U}}) \| p_{\theta_\circ}(\boldsymbol{\mathcal{U}}|\mathbf{y})] + \mathrm{KL}[q_\Phi(\boldsymbol{\mathcal{U}}) \| q_{\Phi_\circ}(\boldsymbol{\mathcal{U}})] \\
&= \log p_{\theta_\circ}(\mathbf{y}) \ .
\end{aligned}
$$

So, **player 2** receives a constant payoff (i.e., independent of $\{\Phi, \theta_\circ\}$) and cannot improve its strategy to achieve a better payoff. Since every player cannot improve

strategy to achieve a better payoff, $(\{\Psi_\circ\}, \{\theta_\circ, \Phi_\circ\})$ is a Nash Equilibrium.

The rest of the proof is similar to that of Proposition 2. □

C.4 Additional Details for Experiments

C.4.1 Synthetic Experiment: Learning a Multi-Modal Posterior Belief

The prior belief is set as a mixture of 5 Gaussians:

$$p(\mathbf{f}) \triangleq p_i \sum_{i=1}^{5} \mathcal{N}(\mu_i \exp(-8x^2), \mathbf{K_{XX}})$$

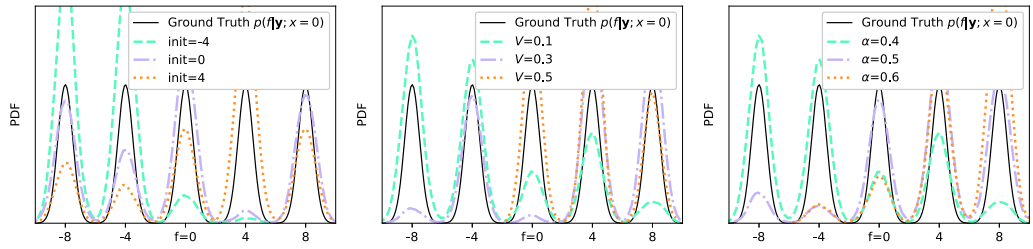
where $p_i \triangleq 1/5$ for $i = 1, \ldots, 5$, $\mu_1 \triangleq -8$, $\mu_2 \triangleq -4$, $\mu_3 \triangleq 0$, $\mu_4 \triangleq 4$, $\mu_5 \triangleq 8$, and $\mathbf{K_{XX}}$ denotes a constant covariance matrix with a constant kernel $k(x, x') \triangleq \sigma_A^2$ and $\sigma_A^2 \triangleq 1/(4 - \exp(-8))$.

Also, $p(\mathbf{y}|\mathbf{f}) = \prod_n p(y_n|f_n) = \prod_n (1/(\sqrt{2\pi}\sigma_B)) \exp(-(y_i - f_i)^2/(2\sigma_B^2))$ with a large noise variance $\sigma_B^2 = 7\exp(8)$. Then, the ground-truth posterior belief with 5 modes can be recovered analytically using Bayes rule:

$$p(\mathbf{f}|\mathbf{y}) = p_i' \sum_{i=1}^{5} \mathcal{N}(\mu_i \exp(-8x^2) + \delta_i, \mathbf{K'_{XX}})$$

where $p_1' = 0.1988$, $p_2' = 0.2004$, $p_3' = 0.2016$, $p_4' = 0.2004$, $p_5' = 0.1988$, $\delta_1 = 0.000479$, $\delta_2 = 0.00024$, $\delta_3 = 0$, $\delta_4 = -0.00024$, $\delta_5 = -0.000479$, and $\mathbf{K'_{XX}}$ denotes a constant covariance matrix with a constant kernel $k'(x, x') \triangleq \sigma_C^2$ and $\sigma_C^2 = 1/4$.

In our implementation, the ground-truth GP kernel hyperparameter values are known to IPVI and SGHMC. We adopt a single inducing input fixed at $z = 0$. The multi-modal posterior belief $p(f|\mathbf{y}; x = 0)$ is then approximated using the samples



(a) (b) (c)

Figure C.1: SGHMC with different hyperparameter settings of learning rate η, momentum $1 - \alpha$, Fisher information V, and initialization init for starting the sampler: (a) $\eta = 0.3, \alpha = 0.4, V = 0.1$; (b) $\eta = 0.3, \text{init} = 4, \alpha = 0.4$; and (c) $\eta = 0.3, \text{init} = 4, V = 0.1$.

from $p(u|\mathbf{y}; z = 0)$.

We vary the number of hidden layers and number of neurons in each hidden layer to obtain generators with different number of parameters in Fig. 4.2c.

In Figure C.1, we give additional results for different hyperparameter setting of SGHMC to show that it is likely to obtain a biased posterior belief.

C.4.2 Experimental Setting for Supervised Learning

Learning Rates. We adopt the default settings of the learning rates of the tested methods from their publicly available implementations. The learning rates and maximum iteration for IPVI are tuned through grid search and cross validation with a default setting of $\alpha_\Psi = 0.05$, $\alpha_\Phi = 0.001$, $\alpha_\theta = 0.025$ and cut-off at a maximum of 20000 iterations. The learning rates for classification is simply set to be 0.02 for all parameters.

Hidden Dimensions. The dimension of inducing variables for all implementations are set to be (a) the same as input dimension for the UCI benchmark regression and Airline datasets, (b) 16 for the YearMSD dataset, and (c) 98 for the classification tasks.

Mini-Batch Sizes. The mini-batch sizes for all implementations are set to be (a) 10000 for the UCI benchmark regression tasks, (b) 20000 for the large-scale regression tasks, and (c) 256 for the classification tasks.

Generator/Discriminator Details. We will describe here the neural network represented by g_{ϕ_ℓ}. Firstly, the noise ϵ has the same dimension as the inputs $\mathbf{X}$ of the dataset. We implement g_{ϕ_ℓ} using a two-layer neural network with hidden dimension being equal to the dimension of $\mathbf{Z}_\ell$ and leaky ReLU activation in the middle. Similarly, we implement T_{ψ_ℓ} using a two-layer neural network with hidden dimension being equal to the dimension of $\mathbf{Z}_\ell$ and leaky ReLU activation in the middle. The network initialization follows random normal distribution.

Mean Function of DGP. The 'skip-layer' connections are implemented in both SGHMC [Havasi *et al.*, 2018] and DSVI [Salimbeni and Deisenroth, 2017] for DGPs and in our IPVI framework as well. The work of [Duvenaud *et al.*, 2014] has analyzed that using a zero mean function in the DGP prior causes some difficulty as each GP mapping is highly non-injective. To mitigate this issue, the work of [Salimbeni and Deisenroth, 2017] has proposed to include a linear mean function $m(\mathbf{X}) = \mathbf{W}\mathbf{X}$ for all hidden layers. The 'skip-layer' connection $\mathbf{W}$ is set to be an identity matrix if the input dimension equals to the output dimension. Otherwise, $\mathbf{W}$ is computed from the top H eigenvectors of the data under SVD. We follow the same setting as this 'skip-layer' mean function. Note that this 'skip-layer' mean function contains no trainable parameters.

Convolutional Skip-layer Connection (CSC). $\mathbf{W}$ is convolution kernel with height and width of 3×3. Note that in CSC, $\mathbf{W}$ is trainable.

Likelihood. For the classification tasks, we use the robust-max multiclass likelihood [Hernández-Lobato *et al.*, 2011]. Tricks like data augmentation are not applied, which means that the accuracy can still be improved further with those additional tricks.

C.4.3 Unsupervised Learning: FreyFace Reconstruction

The dimensions of the hidden layers are 2 for $\mathbf{X}$ and 100 for $\mathbf{F}_1$ for FreyFace Reconstruction. We did not exploit inducing variables here. So, the training is a full DGP. We use PCA as the mean function for this unsupervised learning task.

Reconstruction. Given a trained DGP model, the reconstruction task of a partially observed $\mathbf{y}^\star_O$ is to recover the missing part $\mathbf{y}^\star_U$ such that $\mathbf{y}^\star = [\mathbf{y}^\star_O, \mathbf{y}^\star_U]$. This reconstruction task involves two steps. The first step is to cast it as an DGP inference problem to get the posterior $p(\mathbf{x}^\star|\mathbf{y}^\star_O)$ with a Gaussian likelihood $p(\mathbf{y}^\star_O|\mathbf{y}^\star)$. The second step samples $\mathbf{y}^\star$ from $p(\mathbf{y}^\star|\mathbf{y}^\star_O) = \int p(\mathbf{y}^\star|\mathbf{x}^\star)\ p(\mathbf{x}^\star|\mathbf{y}^\star_O)\ \mathrm{d}\mathbf{x}^\star$.

Appendix D

Appendix for Chapter 5

D.1 Energy Function Details

As illustrated in Figure D.1, the potential function $U(\mathbf{z})$ for each of these four unnormalized distribution is:

(a) $U(\mathbf{z}) = \frac{1}{2}\left(\frac{||\mathbf{z}|| - 2}{0.4}\right) - \ln\left(\exp\left(-\frac{1}{2}(\frac{\mathbf{z}_1 - 2}{0.6})^2 + \exp -\frac{1}{2}(\frac{\mathbf{z}_1 + 2}{0.6})^2\right)\right)$

(b) $U(\mathbf{z}) = -\ln\left(\exp\left(-\frac{1}{2}(\frac{\mathbf{z}_2 - w_1(\mathbf{z})}{0.4})^2\right) + \exp\left(-\frac{1}{2}(\frac{\mathbf{z}_2 - w_1(\mathbf{z}) + w_3(\mathbf{z})}{0.35})^2\right)\right)$

(c) $U(\mathbf{z}) = -\ln\left(\exp\left(-\frac{1}{2}(\frac{\mathbf{z}_2 - w_1(\mathbf{z})}{0.35})^2\right) + \exp\left(-\frac{1}{2}(\frac{\mathbf{z}_2 - w_1(\mathbf{z}) + w_2(\mathbf{z})}{0.35})^2\right)\right)$

(d) $U(\mathbf{z}) = \frac{1}{2}\left(\frac{\mathbf{z}_2 - w_1(\mathbf{z})}{0.4}\right)^2$

where

- $w_1(\mathbf{z}) = \sin(\frac{2\pi\mathbf{z}_1}{4})$ and $w_2(\mathbf{z}) = 3\exp\left(-\frac{1}{2}\left(\frac{\mathbf{z}_1 - 1}{0.6}\right)^2\right)$
- $w_3(\mathbf{z}) = 3\sigma(\frac{\mathbf{z}_1 - 1}{0.3})$ and $\sigma(z) = \frac{1}{1 + \exp(-z)}$

Note that $\mathbf{z}_1$ and $\mathbf{z}_2$ represent the first and second dimension of $\mathbf{z}$ respectively.

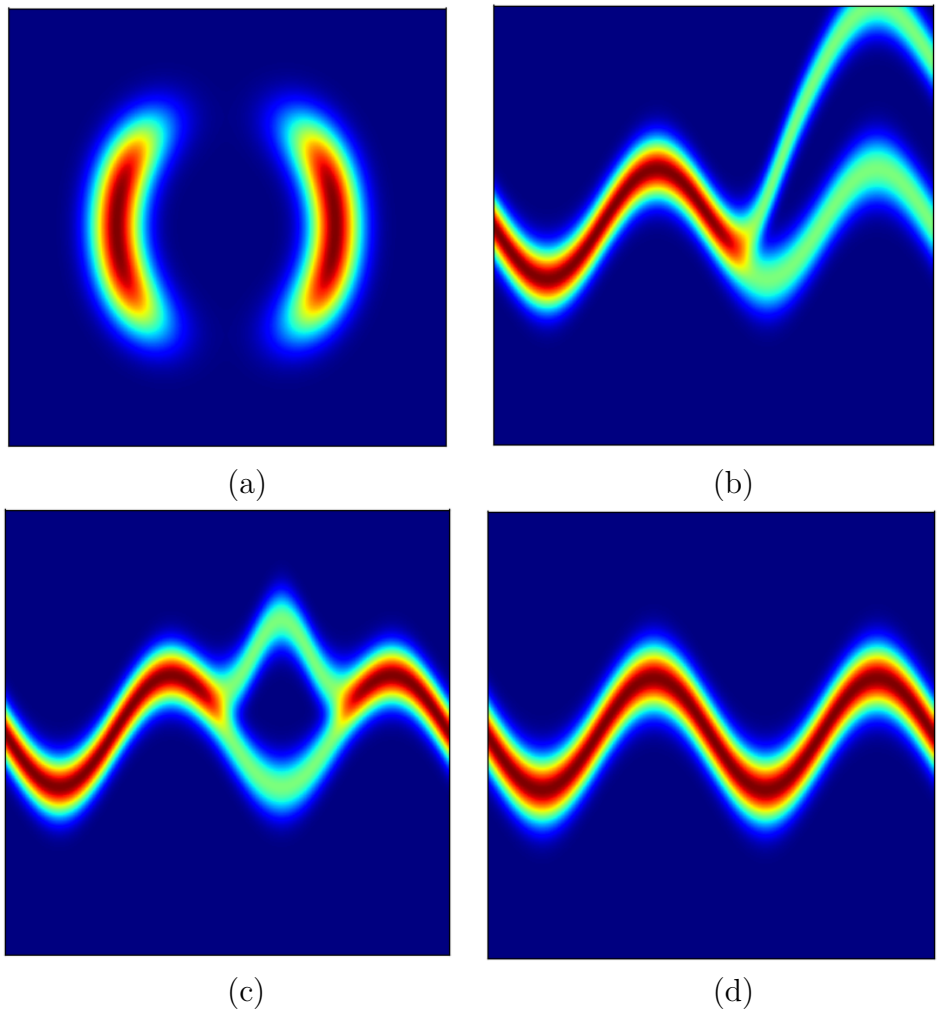


Figure D.1: Four unnormalized distributions

D.2 Additional Details for Experiments

D.2.1 Experimental Setting

Hidden Dimensions. The dimension of inducing variables for all implementations are set to be (a) the same as input dimension for the UCI benchmark regression and Airline datasets, (b) 16 for the YearMSD dataset, and (c) 98 for the classification tasks.

Mini-Batch Sizes. The mini-batch sizes for all implementations are set to be (a) 10000 for the UCI benchmark regression tasks, (b) 20000 for the large-scale regression tasks, and (c) 256 for the classification tasks.

Likelihood. For the classification tasks, we use the robust-max multiclass likelihood [Hernández-Lobato *et al.*, 2011]. Tricks like data augmentation are not applied, which means that the accuracy can still be improved further with those additional tricks.

NF Details. We implement φ_ℓ using a two-layer neural network with hidden dimension being 256 and leaky ReLU activation in the middle. Note that it utilizes separate set of parameters for different layer ℓ.

Mean Function of DGP. We adopt the same setting as that of IPVI for regression and classification.

Bibliography

[Alvarez and Lawrence, 2009] Mauricio Alvarez and Neil D Lawrence. Sparse convolved Gaussian processes for multi-output regression. In *Proc. NeurIPS*, pages 57–64, 2009.

[Álvarez and Lawrence, 2011] Mauricio A Álvarez and Neil D Lawrence. Computationally efficient convolved multiple output Gaussian processes. *The Journal of Machine Learning Research*, 12:1459–1500, 2011.

[Arjovsky *et al.*, 2017] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proc. ICML*, pages 214–223, 2017.

[Awerbuch *et al.*, 2008] Baruch Awerbuch, Yossi Azar, Amir Epstein, Vahab Seyed Mirrokni, and Alexander Skopalik. Fast convergence to nearly optimal solutions in potential games. In *Proc. ACM EC*, pages 264–273, 2008.

[Bauer *et al.*, 2016] Matthias Bauer, Mark van der Wilk, and Carl Edward Rasmussen. Understanding probabilistic sparse Gaussian process approximations. In *Proc. NeurIPS*, pages 1533–1541, 2016.

[Bishop, 2006] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[Bogachev *et al.*, 2005] Vladimir Igorevich Bogachev, Aleksandr Viktorovich Kolesnikov, and Kirill Vladimirovich Medvedev. Triangular transformations of measures. *Sbornik: Mathematics*, 196(3):309, 2005.

[Bui *et al.*, 2016] Thang Bui, Daniel Hernández-Lobato, Jose Hernandez-Lobato, Yingzhen Li, and Richard Turner. Deep Gaussian processes for regression using approximate expectation propagation. In *Proc. ICML*, pages 1472–1481, 2016.

[Cao *et al.*, 2013] Nannan Cao, Kian Hsiang Low, and John M Dolan. Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms. In *Proc. AAMAS*, pages 7–14, 2013.

[Chen *et al.*, 2013] Jie Chen, Nannan Cao, Kian Hsiang Low, Ruofei Ouyang, Colin Keng-Yan Tan, and Patrick Jaillet. Parallel Gaussian process regression with low-rank covariance matrix approximations. In *Proc. UAI*, pages 152–161, 2013.

[Csató and Opper, 2002] Lehel Csató and Manfred Opper. Sparse online Gaussian processes. *Neural Computation*, 14:641–669, 2002.

[Cutajar *et al.*, 2017] Kurt Cutajar, Edwin V Bonilla, Pietro Michiardi, and Maurizio Filippone. Random feature expansions for deep Gaussian processes. In *Proc. ICML*, pages 884–893, 2017.

[Dai *et al.*, 2016] Zhenwen Dai, Andreas Damianou, Javier González, and Neil Lawrence. Variational auto-encoded deep Gaussian processes. In *Proc. ICLR*, 2016.

[Damianou and Lawrence, 2013] Andreas Damianou and Neil Lawrence. Deep Gaussian processes. In *Proc. AISTATS*, pages 207–215, 2013.

[Daxberger and Low, 2017] Erik A Daxberger and Kian Hsiang Low. Distributed batch Gaussian process optimization. In *Proc. ICML*, pages 951–960, 2017.

[De G. Matthews *et al.*, 2017] Alexander G De G. Matthews, Mark Van Der Wilk, Tom Nickson, Keisuke Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. GPflow: A Gaussian process library using TensorFlow. *The Journal of Machine Learning Research*, 18(1):1299–1304, 2017.

[Deisenroth and Ng, 2015] Marc Peter Deisenroth and Jun Wei Ng. Distributed Gaussian processes. In *Proc. ICML*, pages 1481–1490, 2015.

[Dinh and Bengio, 2016] Laurent Dinh and Samy Bengio. Density estimation using real nvp. In *Proc. ICLR*, 2016.

[Duvenaud *et al.*, 2014] David Duvenaud, Oren Rippel, Ryan Adams, and Zoubin Ghahramani. Avoiding pathologies in very deep networks. In *Proc. AISTATS*, pages 202–210, 2014.

[Gal and Turner, 2015] Yarin Gal and Richard Turner. Improving the Gaussian process sparse spectrum approximation by representing uncertainty in frequency inputs. In *Proc. ICML*, pages 655–664, 2015.

[Gal and van der Wilk, 2014] Yarin Gal and Mark van der Wilk. Variational inference in sparse Gaussian process regression and latent variable models–a gentle tutorial. arXiv:1402.1412, 2014.

[Gal *et al.*, 2014] Yarin Gal, Mark Van Der Wilk, and Carl Edward Rasmussen. Distributed variational inference in sparse Gaussian process regression and latent variable models. In *Proc. NeurIPS*, pages 3257–3265, 2014.

[Garriga-Alonso *et al.*, 2019] Adrià Garriga-Alonso, Carl Edward Rasmussen, and Laurence Aitchison. Deep convolutional networks as shallow Gaussian processes. In *Proc. ICLR*, 2019.

[Genevay *et al.*, 2018] Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with Sinkhorn divergences. In *Proc. AISTATS*, pages 1608–1617, 2018.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. NeurIPS*, pages 2672–2680, 2014.

[Goodfellow, 2016] Ian Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. arXiv:1701.00160, 2016.

[Gulrajani *et al.*, 2017] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein gans. In *Proc. NeurIPS*, pages 5767–5777, 2017.

[Havasi *et al.*, 2018] Marton Havasi, José Miguel Hernández-Lobato, and Juan José Murillo-Fuentes. Inference in deep Gaussian processes using stochastic gradient Hamiltonian Monte Carlo. In *Proc. NeurIPS*, pages 7517–7527, 2018.

[Hensman and Lawrence, 2014] James Hensman and Neil D. Lawrence. Nested variational compression in deep Gaussian processes. arXiv:1412.1370, 2014.

[Hensman *et al.*, 2013] James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. In *Proc. UAI*, pages 282–290, 2013.

[Hernández-Lobato *et al.*, 2011] Daniel Hernández-Lobato, José M Hernández-Lobato, and Pierre Dupont. Robust multi-class Gaussian process classification. In *Proc. NeurIPS*, pages 280–288, 2011.

[Hernández-Lobato *et al.*, 2014] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Proc. NeurIPS*, pages 918–926, 2014.

[Hoang *et al.*, 2014] Trong Nghia Hoang, Kian Hsiang Low, Patrick Jaillet, and Mohan Kankanhalli. Nonmyopic ϵ-Bayes-optimal active learning of Gaussian processes. In *Proc. ICML*, pages 739–747, 2014.

[Hoang *et al.*, 2015] Trong Nghia Hoang, Quang Minh Hoang, and Kian Hsiang Low. A unifying framework of anytime sparse Gaussian process regression models with stochastic variational inference for big data. In *Proc. ICML*, pages 569–578, 2015.

[Hoang *et al.*, 2016] Trong Nghia Hoang, Quang Minh Hoang, and Kian Hsiang Low. A distributed variational inference framework for unifying parallel sparse Gaussian process regression models. In *Proc. ICML*, pages 382–391, 2016.

[Hoang *et al.*, 2017] Quang Minh Hoang, Trong Nghia Hoang, and Kian Hsiang Low. A generalized stochastic variational Bayesian hyperparameter learning framework for sparse spectrum Gaussian process regression. In *Proc. AAAI*, pages 2007–2014, 2017.

[Hoang *et al.*, 2018] Trong Nghia Hoang, Quang Minh Hoang, and Kian Hsiang Low. Decentralized high-dimensional Bayesian optimization with factor graphs. In *Proc. AAAI*, pages 3231–3238, 2018.

[Hoffman *et al.*, 2013] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[Hornik *et al.*, 1989] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[Huszár, 2017] Ferenc Huszár. Variational inference using implicit distributions. arXiv:1702.08235, 2017.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*, pages 448–456, 2015.

[Jacot *et al.*, 2018] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Proc. NeurIPS*, pages 8571–8580, 2018.

[Karras *et al.*, 2018] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *Proc. ICLR*, 2018.

[Kendall and Gal, 2017] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Proc. NeurIPS*, pages 5574–5584, 2017.

[Khan *et al.*, 2018] Mohammad Emtiyaz Khan, Didrik Nielsen, Voot Tangkaratt, Wu Lin, Yarin Gal, and Akash Srivastava. Fast and scalable bayesian deep learning by weight-perturbation in adam. In *Proc. ICML*, pages 2616–2625, 2018.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv:1412.6980, 2014.

[Kingma and Dhariwal, 2018] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Proc. ICML*, pages 10215–10224, 2018.

[Kingma and Welling, 2013] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *Proc. ICLR*, 2013.

[Kurakin *et al.*, 2016] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *Proc. ICLR*, 2016.

[Lawrence, 2004] Neil D Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Proc. NeurIPS*, pages 329–336, 2004.

[LeCun *et al.*, 2015] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[Lee *et al.*, 2018] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as Gaussian processes. In *Proc. ICLR*, 2018.

[Li and Gal, 2017] Yingzhen Li and Yarin Gal. Dropout inference in Bayesian neural networks with alpha-divergences. In *Proc. ICML*, pages 2052–2061, 2017.

[Ling *et al.*, 2016] Chun Kai Ling, Kian Hsiang Low, and Patrick Jaillet. Gaussian process planning with Lipschitz continuous reward functions: Towards unifying Bayesian optimization, active learning, and beyond. In *Proc. AAAI*, pages 1860–1866, 2016.

[Low *et al.*, 2008] Kian Hsiang Low, John M Dolan, and Pradeep Khosla. Adaptive multi-robot wide-area exploration and mapping. In *Proc. AAMAS*, pages 23–30, 2008.

[Low *et al.*, 2009] Kian Hsiang Low, John M Dolan, and Pradeep Khosla. Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing. In *Proc. ICAPS*, pages 233–240, 2009.

[Low *et al.*, 2011] Kian Hsiang Low, John M Dolan, and Pradeep Khosla. Active Markov information-theoretic path planning for robotic environmental sensing. In *Proc. AAMAS*, pages 753–760, 2011.

[Matthews *et al.*, 2018] Alexander G de G Matthews, Mark Rowland, Jiri Hron, Richard E Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. In *Proc. ICLR*, 2018.

[Medvedev, 2008] Kirill Vladimirovich Medvedev. Certain properties of triangular transformations of measures. *Theory of Stochastic Processes*, 14(1):95–99, 2008.

[Mescheder *et al.*, 2017] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks. In *Proc. ICML*, pages 2391–2400, 2017.

[Minka, 2001] Thomas P Minka. Expectation propagation for approximate Bayesian inference. In *Proc. UAI*, pages 362–369, 2001.

[Montavon *et al.*, 2016] Grégoire Montavon, Klaus-Robert Müller, and Marco Cuturi. Wasserstein training of restricted Boltzmann machines. In *Proc. NeurIPS*, pages 3718–3726, 2016.

[Neal, 1995] Radford M Neal. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995.

[Nguyen *et al.*, 2014] Trung V Nguyen, Edwin V Bonilla, et al. Collaborative multi-output Gaussian processes. In *Proc. UAI*, pages 643–652, 2014.

[Novak *et al.*, 2019] Roman Novak, Lechao Xiao, Yasaman Bahri, Jaehoon Lee, Greg Yang, Jiri Hron, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Bayesian deep convolutional networks with many channels are Gaussian processes. In *Proc. ICLR*, 2019.

[Ouyang *et al.*, 2014] Ruofei Ouyang, Kian Hsiang Low, Jie Chen, and Patrick Jaillet. Multi-robot active sensing of non-stationary Gaussian process-based environmental phenomena. In *Proc. AAMAS*, pages 573–580, 2014.

[Petersen *et al.*, 2012] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. Technical report, 2012.

[Peyré *et al.*, 2019] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

[Quiñonero-Candela and Rasmussen, 2005] Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.

[Rasmussen and Williams, 2006] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[Rezende and Mohamed, 2015] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proc. ICML*, pages 1530–1538, 2015.

[Roweis *et al.*, 2002] Sam T Roweis, Lawrence K Saul, and Geoffrey E Hinton. Global coordination of local linear models. In *Proc. NeurIPS*, pages 889–896, 2002.

[Salimans *et al.*, 2016] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Proc. NeurIPS*, pages 2234–2242, 2016.

[Salimbeni and Deisenroth, 2017] Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep Gaussian processes. In *Proc. NeurIPS*, pages 4588–4599, 2017.

[Salimbeni *et al.*, 2018] Hugh Salimbeni, Stefanos Eleftheriadis, and James Hensman. Natural gradients in practice: Non-conjugate variational inference in Gaussian process models. In *Proc. AISTATS*, pages 689–697, 2018.

[Seeger *et al.*, 2003] Matthias Seeger, Christopher Williams, and Neil Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *Proc. AISTATS*, 2003.

[Snelson and Ghahramani, 2005] Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Proc. NeurIPS*, pages 1257–1264, 2005.

[Snelson and Ghahramani, 2007] Edward Snelson and Zoubin Ghahramani. Local and global sparse Gaussian process approximations. In *Proc. AISTATS*, 2007.

[Springenberg *et al.*, 2016] Jost Tobias Springenberg, Aaron Klein, Stefan Falkner, and Frank Hutter. Bayesian optimization with robust Bayesian neural networks. In *Proc. NeurIPS*, pages 4134–4142, 2016.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[Tabak and Turner, 2013] Esteban G Tabak and Cristina V Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013.

[Tabak and Vanden-Eijnden, 2010] Esteban G Tabak and Eric Vanden-Eijnden. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, 2010.

[Titsias and Lázaro-Gredilla, 2013] Michalis Titsias and Miguel Lázaro-Gredilla. Variational inference for Mahalanobis distance metrics in Gaussian process regression. In *Proc. NeurIPS*, pages 279–287, 2013.

[Titsias, 2009] Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Proc. AISTATS*, pages 567–574, 2009.

[van den Oord *et al.*, 2016] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. arXiv:1609.03499, 2016.

[van der Wilk *et al.*, 2019] Mark van der Wilk, ST John, Artem Artemev, and James Hensman. Variational Gaussian process models without matrix inverses. Technical report, 2019.

[Villani, 2003] Cédric Villani. *Topics in optimal transportation*. Number 58. American Mathematical Society, 2003.

[Wainwright and Jordan, 2008] Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

[Wang *et al.*, 2018] Kuan-Chieh Wang, Paul Vicol, James Lucas, Li Gu, Roger Grosse, and Richard Zemel. Adversarial distillation of Bayesian neural network posteriors. In *Proc. ICML*, pages 5177–5186, 2018.

[Wang *et al.*, 2019] Ke Wang, Geoff Pleiss, Jacob Gardner, Stephen Tyree, Kilian Q Weinberger, and Andrew Gordon Wilson. Exact Gaussian processes on a million data points. In *Proc. NeurIPS*, pages 14622–14632, 2019.

[Yu *et al.*, 2019a] Haibin Yu, Yizhou Chen, Zhongxiang Dai, Kian Hsiang Low, and Patrick Jaillet. Implicit posterior variational inference for deep Gaussian processes. In *Proc. NeurIPS*, pages 14475–14486, 2019.

[Yu *et al.*, 2019b] Haibin Yu, Trong Nghia Hoang, Kian Hsiang Low, and Patrick Jaillet. Stochastic variational inference for Bayesian sparse Gaussian process regression. In *Proc. IJCNN*, pages 1–8, 2019.

[Zhang *et al.*, 2016] Yehong Zhang, Trong Nghia Hoang, Kian Hsiang Low, and Mohan Kankanhalli. Near-optimal active learning of multi-output Gaussian processes. In *Proc. AAAI*, pages 2351–2357, 2016.

[Zhang *et al.*, 2019] Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient MCMC for Bayesian deep learning. arXiv:1902.03932, 2019.