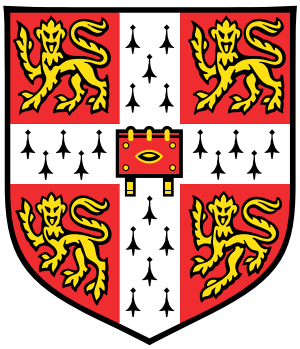
Spatio-Temporal Variational Autoencoders



Matthew Christopher Ashman

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy in Machine Learning and Machine Intelligence

St John's College

August 2020

In memory of Sam Fitzsimmons

Declaration

I, Matthew Christopher Ashman of St John's College, being a candidate for the M.Phil. in Machine Learning and Machine Intelligence, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

The code used in this thesis does not rely on any previously written software other than the standard Python packages.

Word count: 14610.

Matthew Christopher Ashman
August 2020

Acknowledgements

I am enormously grateful to have been supervised by Rich Turner, whose guidance and technical insights have been invaluable. Thank you for your countless ideas, for introducing me to researchers of the highest calibre, and for establishing an academic environment which I have enjoyed and benefited from tremendously.

I owe sincere thanks to Jonathan So, whose clarity of thought helped me overcome many of the technical hurdles I faced. I am fortunate enough to have had many illuminating discussions with Vincent Fortuin, Will Tebbutt and Wessel Bruinsma. Thank you for sharing your ideas and being generous with your time.

Thank you to the MLMI cohort. Although our year was cut short, I hope it planted the seeds of a lasting friendship.

Finally, this, nor any of my academic achievements, would have been possible without the love and support of my parents, Chris and Joanne Ashman, and partner, Penelope Jones.

Abstract

The ubiquitous interest in deep latent variable models within the machine learning community has been fuelled by the development of the variational autoencoder (VAE). VAEs provide a framework for performing fast, scalable inference in deep latent variable models, facilitating their deployment on the large, multi-dimensional and richly structured datasets omnipresent in modern science and engineering. Increasingly large volumes of such data that also exhibit strong dependencies across space and time are arising from a wealth of domains, including environmental, social and earth sciences. Crucial to the advancement of these fields are the tools to effectively model spatio-temporal data. Despite their wide applicability, VAEs are ill-equipped to model such data. At the crux of this inadequacy is a deficiency in the probabilistic model, specifically, the assumption that observations are independent and identically distributed.

In contrast to VAEs, Gaussian processes (GPs) are an extremely effective tool for modelling data that exhibits strong dependencies. Unfortunately, the power of GPs necessitates an often undermining computational burden - scaling cubically with the number of data points and observed dimensions. This prohibits their application in the large data regime. Furthermore, GPs are comparatively inexpressive relative to 'deep' machine learning models, such as the VAE. Whilst the construction of more expressive GPs is possible, this is typically a hand-crafted process which only adds to the computational complexity, unlike the automatic feature learning intrinsic to deep models.

This thesis seeks to unify the complementary strengths of VAEs and GPs, forming a novel family of VAEs for the effective modelling of spatio-temporal datasets. The amalgamation of the two models is natural; however, it requires careful consideration of approximate inference techniques to ensure the benefits of each are realised. We establish the theoretical framework for achieving this, paying particular attention to the preservation of structure in the approximate posterior and the principled handling of partially observed data. We provide an extension to the sparse GP literature, developing a scalable technique for the introduction of sparse approximations into our family of spatio-temporal VAEs. Finally, we demonstrate state-of-the-art performance relative to existing multi-output GP models and structured VAEs in a variety of experiments involving spatio-temporal datasets.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms / Abbreviations

DLVM Deep Latent Variable Model

DNN Deep Neural Network

DS-PD Doubly-Stochastic Path Derivative

DS-SF Doubly-Stochastic Score Function

EEG Electroencephalogram

ELBO Evidence Lower Bound

EM Expectation Maximisation

EP Expectation Propagation

GMM Gaussian Mixture Model

GPAR Gaussian Process Autoregressive Regression Model

GP Gaussian Process

GP-VAE Gaussian Process Variational Autoencoder

IGP Independent Gaussian Processes

iid Independently and Identically Distributed

KL Kullback-Liebler

LDS Latent Dynamical System

LOTUS Law of Unconscious Statisticians

LVM Latent Variable Model

MAE Mean Absolute Error

MCMC Markov Chain Monte Carlo

MC-VI Monte Carlo Variational Inference

NLL Negative Log-Likelihood

RMSE Root Mean Squared Error

SA-PD Semi-Analytic Path Derivative

SA-SF Semi-Analytic Score Function

SE Squared Exponential

SIN Structured Inference Network

SMSE Standardised Mean Squared Error

SVAE Structured Variational Autoencoder

VAE Variational Autoencoder

VFE Variational Free Energy

VI Variational Inference

ZI Zero Imputation

1 | Introduction

Observed data can be consistent with many explanations, the plausibility of which changes as more data comes to light. Uncertainty in the explanation translates to uncertainty in predictions about unobserved quantities, including future data or underlying state of the world. Accounting for this should form the basis of any principled machine learning model. Probabilistic machine learning addresses the presence of uncertainty at a foundational level. At its core is the treatment of quantities, including the observed data, as random variables related to each other through probability distributions. Once a probabilistic model has been constructed - that is, the joint probability distribution through which the unobserved and observed variables are related - the basic rules of probability are applied to translate the information gained by the observed data into probability distributions over the unobserved data. This process is known as *inference*, and constitutes the main hurdle of most probabilistic machine learning models.

A hallmark of probabilistic modelling is the use of *latent variables*: variables that form an intrinsic part of the probabilistic model, but are never directly observed. More often than not, latent variables are introduced to represent simple explanations of more complex observations. Performing inference over latent variables forms the basis of unsupervised learning, a cornerstone of machine learning. The probability distribution that relates latent variables to the observed quantities is termed the *likelihood*. Used together with the *prior* distribution over latent variables, the complete probabilistic model is defined. The effectiveness of a latent variable model is determined by its ability to model the properties of the data necessary for the prediction task of interest. One approach to this is the development of flexible likelihoods, the impetus for which is the nature of datasets emerging from the modern world of which many are extremely large and high dimensional. Flexible likelihoods are a necessity for modelling multi-dimensional observations, whereas richly structured priors often impose an overwhelming computational burden for large datasets.

An immediate solution to the design of flexible likelihoods is the use of deep neural networks (DNNs) to parameterise the likelihood function. Unfortunately, performing inference in the corresponding deep latent variable models (DLVMs) is intractable. One of the most significant developments in the field of probabilistic modelling is the variational autoencoder (VAE)

(Kingma and Welling, 2014). VAEs provide a framework for performing fast and scalable approximate inference in DLVMs, facilitating their deployment on large, multi-dimensional and richly structured datasets. Yet, increasingly large volumes of such data that also exhibit strong spatio-temporal dependencies are arising from a wealth of domains, including environmental, social and earth sciences (Atluri et al., 2018). Crucial to the advancement of these fields are the tools to effectively model spatio-temporal data. Despite their wide applicability, standard VAEs are ineffective in achieving this. At the crux of this inadequacy is a deficiency in the probabilistic model typically employed, specifically, the assumption that observations are independent from one another. This removes any capacity to model dependencies between observations, rendering it unable to explain spatio-temporally distributed data.

An alternative approach to incorporating flexibility into latent variable models is through the use of nonparametric prior distributions (Ghahramani, 2015). In particular, Gaussian processes (GPs) are a flexible nonparametric model for describing probability distributions over latent functions. Properties such as smoothness, stationarity or periodicity across the input domain are incorporated into the covariance function, making GPs a natural fit to data distributed across space and time. Inherent to GPs, and more generally Bayesian nonparametric models, is the ability to provide principled uncertainty estimates and robustness in settings in which the complexity of data is unknown. Unfortunately, the power of GPs necessitates an often undermining computational burden, scaling cubically with both the number of data points and observed dimensions. This prohibits their application in the large data regime. Moreover, the flexibility of GPs often relies on hand-crafted covariance functions which only adds to the computational complexity (Duvenaud, 2014). This contrasts with the automatic hierarchical feature learning intrinsic to deep neural networks, and in turn DLVMs (Bengio et al., 2013; Hinton, 2007).

In this thesis, we seek to unify the complementary strengths of VAEs and GPs. The amalgamation of the two models is natural: we simply place a GP prior over the latent variables of a DLVM. However, this requires careful consideration of approximate inference to ensure the benefits of each are realised. We present a theoretical framework for this unification and demonstrate the effectiveness of the resultant model on a variety of experiments involving spatio-temporal datasets.

1.1 Thesis Overview

The structure of this thesis is as follows:

Chapter 2 establishes the theoretical backdrop upon which the developments in this thesis build. Alongside the introduction of VAEs and GPs, we present the use of variational

inference for approximating inference in latent variable models as well as the recently developed ideas behind Deep Sets.

Chapter 3 introduces a novel family of spatio-temporal VAEs - the GP-VAE - laying out the framework through which learning and inference can be performed efficiently using the theories established in Chapter 2. We pay special consideration to the presence of partially observed data, providing a general framework for constructing theoretically principled inference networks capable of handling missing values. We outline the use of sparse GP approximations in the GP-VAE, marking an important development in the existing sparse GP literature that extends well beyond their current capability.

Chapter 4 reviews the related literature, providing a unifying connection between our model and other multi-output GPs. Notably, we demonstrate that the GP-VAE is a special case of Johnson et al.'s (2016) structured variational autoencoder with a GP prior.

Chapter 5 evaluates the empirical performance of the GP-VAE on a number of experiments involving spatio-temporal datasets of distinguishable characteristics. We demonstrate the comparatively superior performance of the GP-VAE relative to other multi-output GPs, other structured VAEs and the standard VAE. Most importantly, we show that the sparse GP-VAE scales effectively to large datasets, opening the door to a wealth of future research.

Chapter 6 provides a summary of the contributions of this thesis and sheds light on what we believe to be the most promising avenues for future research.

2 | Background Theory

This chapter presents the theoretical underpinnings of the models developed in the remainder of the thesis. Section 2.1 begins with an introduction of latent variable models, leading into an outline of variational inference in Section 2.2. Section 2.3 utilises the theory set out in the preceding two sections, presenting the ideas central to the development of VAEs. Section 2.4 details Gaussian process models, their application to regression problems and the use of sparse Gaussian processes. Finally, Section 2.5 explores the recently developed ideas behind constructing permutation invariant set functions, in particular those of Deep Sets.

2.1 Latent Variable Models

A widely used approach to increasing the effectiveness of probabilistic models is to augment the observed variables with an additional set of latent, or hidden, variables (Jordan, 1998). Rather than specifying the distribution over the observed variables alone, latent variable models (LVMs) define a joint distribution over the augmented space:

$$p_\theta(\mathbf{y}, \mathbf{z}) = p_\theta(\mathbf{y}|\mathbf{z})p_\theta(\mathbf{z}), \tag{2.1}$$

where θ denotes the set of model parameters, $p_\theta(\mathbf{z})$ the prior distribution and $p_\theta(\mathbf{y}|\mathbf{z})$ the likelihood of the latent variables[1]. The distribution over the observed variables is recovered by integrating out the latent variables:

$$p_\theta(\mathbf{y}) = \int p_\theta(\mathbf{y}|\mathbf{z})p_\theta(\mathbf{z})d\mathbf{z}. \tag{2.2}$$

[1]Strictly speaking, there is no philosophical difference between latent variables and model parameters, as model parameters are also unobserved variables that we wish to infer. In practice, however, latent variables tend to be *local* - typically having a set of latent variables associated with each observation, or group of observations. In contrast, model parameters tend to be *global* - typically being shared across all observed and unobserved variables. Furthermore, LVMs rarely perform fully Bayesian inference over the model parameters, which are often treated as deterministic values to be found through maximum likelihood or maximum a posteriori estimation.

The principal advantage of LVMs is that, despite $p_\theta(\mathbf{y}|\mathbf{z})$ and $p_\theta(\mathbf{z})$ often being relatively simple, the marginal distribution $p_\theta(\mathbf{y})$ can be extremely complex[2]. Many of the most popular probabilistic models - including the family of linear Gaussian models, the Gaussian mixture model and latent Dirichlet allocation (Blei et al., 2003; Roweis and Ghahramani, 1999) - use latent variables to model data with greater effect. For settings in which the generative process of the observed data is known a priori, latent variables offer a route through which this prior knowledge can be incorporated into the probabilistic model. In the general case, however, it is common to place vague prior distributions over the latent variables in conjunction with flexible likelihoods. Doing so aids the discovery of simple, low-dimensional representations of data, forming the bedrock of unsupervised learning. Indeed, a prominent motivation for LVMs is their use in dimensionality reduction and unsupervised representation learning, early examples of which include probabilistic principal component analysis (Tipping and Bishop, 1999) and the Gaussian process latent variable model (Lawrence, 2004).

The focus of this thesis is on LVMs with a continuous latent variable, $\mathbf{z}_n$, associated with each observation $\mathbf{y}_n$. Such models typically assume a directed generative process in which observations are conditionally independent of each other given their corresponding latent variables:

$$\begin{aligned} \mathbf{z} &\sim p_\theta(\mathbf{z}) \\ \mathbf{y}|\mathbf{z} &\sim \prod_{n=1}^{N} p_\theta(\mathbf{y}_n|\mathbf{z}_n). \end{aligned} \tag{2.3}$$

Imposing conditional independence between observations enables efficient computations in the model (Bishop, 2006). A frequently used method for introducing flexibility into the likelihood function is to use deep neural networks (DNNs) to map from the latent variable to parameters of the likelihood, often chosen to be either a Bernoulli (or categorical) distribution for discrete observations or a Gaussian distribution for continuous observations. LVMs whose distributions are parameterised by DNNs are known as deep latent variable models (DLVMs).

2.1.1 Learning and Inference in LVMs

The computations of interest in LVMs are learning the model parameters θ and obtaining the posterior distribution over latent variables $p_\theta(\mathbf{z}|\mathbf{y})$. Whilst the fully Bayesian approach would have us specify a prior distribution over θ and compute the posterior distribution $p(\theta|\mathbf{y})$, this is rarely tractable for models of interest[3]. A more pragmatic approach is to find the model

[2] It allows us to express an intractable marginal distribution in terms of a tractable joint distribution.

[3] An alternative option would be to use variational inference to find a joint approximate posterior, $q_\phi(\theta, \mathbf{z})$. However, it could be argued that this would waste computational resources on obtaining a reasonable approximation to $p(\theta|\mathbf{y})$ as $p(\mathbf{z}|\mathbf{y})$ is of greater inferential interest.

parameters which maximise the log marginal likelihood:

$$\theta^* = \arg\max_{\theta} \ \log p_\theta(\mathbf{y}). \tag{2.4}$$

Regrettably, for general LVMs (2.4) has no closed form solution. A widely used alternative is the expectation maximisation (EM) algorithm (Dempster et al., 1977), which iterates between an expectation (E) step and a maximisation (M) step:

E step: Compute the posterior probability for current model parameters θ^τ according to Bayes' rule:

$$p_{\theta^\tau}(\mathbf{z}|\mathbf{y}) = \frac{p_{\theta^\tau}(\mathbf{y}, \mathbf{z})}{p_{\theta^\tau}(\mathbf{y})}.$$

Set $q^\tau(\mathbf{z}) = p_{\theta^\tau}(\mathbf{z}|\mathbf{y})$.

M step: Update the model parameters according to

$$\theta^{\tau+1} = \arg\max_{\theta} \ \mathbb{E}_{q^\tau(\mathbf{z})} \left[\log p_\theta(\mathbf{y}, \mathbf{z})\right].$$

Note that the EM algorithm unifies inference and learning: to perform the E step, we require computation of the posterior $p_\theta(\mathbf{z}|\mathbf{y})$. Yet, for many LVMs of interest - including DLVMs - computing this posterior distribution is infeasible due to the intractability of the marginal likelihood, $p_\theta(\mathbf{y})$. Approximating learning and inference in LVMs constitutes a pillar of modern developments in probabilistic machine learning. Broadly speaking, approximate inference frameworks can be divided into two camps: those using Markov chain Monte Carlo (MCMC) techniques and those using distributional approximations.

Markov Chain Monte Carlo

At the core of MCMC techniques is the Monte Carlo approximation, which approximates expectations under some distribution $p(\mathbf{x})$ as

$$\mathbb{E}_{p(\mathbf{x})}\left[f(\mathbf{x})\right] = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} \ \approx \ \frac{1}{S}\sum_{s=1}^{S} f(\mathbf{x}^s) \quad \text{where} \quad \mathbf{x}^s \sim p(\mathbf{x}). \tag{2.5}$$

The approximation is said to be unbiased if $\mathbb{E}\left[\frac{1}{S}\sum_{s=1}^{S} f(\mathbf{x}^s)\right] = \mathbb{E}_{p(\mathbf{x})}\left[f(\mathbf{x})\right]$. (2.5) assumes the distribution $p(\mathbf{x})$ can be sampled from; yet, for approximating inference, samples must be drawn from the posterior $p_\theta(\mathbf{z}|\mathbf{y})$ which is known to be intractable. Mathematicians have devised a plethora of techniques for drawing samples from simpler, tractable distributions, and then correcting those samples to better approximate expectations over $p_\theta(\mathbf{z}|\mathbf{y})$. MCMC describes a family of such techniques. MCMC methods construct a Markov chain process whose stationary distribution is $p_\theta(\mathbf{z}|\mathbf{y})$ (Gelman et al., 2013). Provided enough samples are drawn

from the Markov chain, they can be used to form unbiased Monte Carlo approximations. Two of the earliest, yet still most widely used, examples of MCMC techniques are the Metropolis-Hastings algorithm (Hastings, 1970) and Gibbs sampling (Geman and Geman, 1984). A notable development is that of Hamiltonian Monte Carlo (Duane et al., 1987; Neal, 1995), which addressed the 'random walk behaviour' observed in other methods and permitted the scalability of MCMC to extremely complex distributions.

Unfortunately, this is often a laboured process. The time taken for MCMC algorithms to converge towards the target distribution can be cripplingly slow. Further, for high-dimensional posterior distributions - which is true for LVMs with local latent variables - the number of samples required to form an acceptable approximation can be extremely large and the simulation time required impractical.

Distributional Approximations

Rather than seeking to form Monte Carlo estimates, distributional approximation techniques construct an approximation to the true posterior, $q(\mathbf{z})$. The approximate posterior is used in place of $p_\theta(\mathbf{z}|\mathbf{y})$ when computing expectations:

$$\begin{aligned} \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{y})}\left[f(\mathbf{z})\right] &= \int p_\theta(\mathbf{z}|\mathbf{y}) f(\mathbf{z}) d\mathbf{z} \\ &\approx \int q(\mathbf{z}) f(\mathbf{z}) d\mathbf{z} = \mathbb{E}_{q(\mathbf{z})}\left[f(\mathbf{z})\right]. \end{aligned} \tag{2.6}$$

By restricting $q(\mathbf{x})$ to be much simpler than $p_\theta(\mathbf{z}|\mathbf{y})$, computing the approximation becomes considerably easier. There exist a number of distributional approximate inference techniques, most notably: the Laplace approximation, popularised by MacKay (1992) for use in Bayesian neural networks; Minka's (2001) expectation propagation, which has been applied to a wide variety of problem domains including Gaussian processes (Bui, 2018; Csató and Opper, 2002; Kuss and Rasmussen, 2005); Opper's (1998) assumed density filtering, a precursor to expectation propagation that has found recent success in Bayesian neural networks (Hernández-Lobato and Adams, 2015); and variational inference, which we shall review in the following section.

2.2 Variational Inference

The variational methodology is to reformulate quantities of interest in terms of finding a solution to an optimisation problem (Jordan et al., 1999; Wainwright and Jordan, 2008). The optimisation problem can then be 'relaxed' to retain tractability. Variational inference (VI) describes the variational approach to approximating probabilistic inference. In particular, finding the quantity of interest - the true posterior distribution $p_\theta(\mathbf{z}|\mathbf{y})$ - can be reformulated

as finding the distribution $q(\mathbf{z})$ which minimises some divergence measure D:

$$q^*(\mathbf{z}) = \arg\min_{q(\mathbf{z})} \; D\left(q(\mathbf{z}) \parallel p(\mathbf{z}|\mathbf{y})\right). \tag{2.7}$$

The divergence measure must satisfy $D\left(q(\mathbf{z}) \parallel p(\mathbf{z}|\mathbf{y})\right)$, with equality if and only if $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{y})$. The optimisation objective of (2.7) can be relaxed by confining $q(\mathbf{z})$ to lie within a tractable family of distributions, typically defined as the set of parametric distributions $q_\phi(\mathbf{z})$ where ϕ are the variational parameters. Whilst it is necessary to restrict the family of approximate distributions to comprise only those that are tractable, the effectiveness of VI hinges on the family being sufficiently flexible to provide a good approximation to the true posterior. A common misconception is that by increasing the flexibility of the approximating family, the approximating distribution is at risk of overfitting. This is incorrect: advancing the optimisation objective only improves the approximation to the desired answer. The distinction between variational and model parameters is of great importance as it permits us to add complexity to the approximate posterior without recasting the modelling assumptions.

There exist a wealth of valid divergence measures, most notably the family of Rényi alpha divergences (Li and Turner, 2016). Traditional VI uses the exclusive Kullback-Liebler (KL) divergence (Kullback and Leibler, 1951), defined as

$$\mathrm{KL}\left(q_\phi(\mathbf{z}) \parallel p_\theta(\mathbf{z}|\mathbf{y})\right) = \int q_\phi(\mathbf{z}) \log \frac{q_\phi(\mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{y})} d\mathbf{z} = \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{y})}\right]. \tag{2.8}$$

In general, $\mathrm{KL}\left(q_\phi(\mathbf{z}) \parallel p_\theta(\mathbf{z}|\mathbf{y})\right) \neq \mathrm{KL}\left(p_\theta(\mathbf{z}|\mathbf{y}) \parallel q_\phi(\mathbf{z})\right)$ - the latter divergence is known as the inclusive KL divergence which is used in expectation propagation. Minimising the exclusive KL divergence favours approximate distributions that are zero in regions where $p_\theta(\mathbf{z}|\mathbf{y})$ is zero[4]. If $q_\phi(\mathbf{z})$ is unimodal (e.g. Gaussian) this results in 'mode matching' or 'zero forcing' behaviour, which contrasts 'mass covering' behaviour encouraged by the inclusive KL divergence. These differences are illustrated in Figure 2.1. An important consequence of this is that the approximate posteriors found using VI tend to underestimate the uncertainty of the true posterior (Turner and Sahani, 2011).

2.2.1 The Evidence Lower Bound

$\mathrm{KL}\left(q_\phi(\mathbf{z}) \parallel p_\theta(\mathbf{z}|\mathbf{y})\right)$ requires computation of $p_\theta(\mathbf{z}|\mathbf{y})$, the very distribution we are seeking to approximate. Thus, direct minimisation of the KL divergence is infeasible. Fortunately, it is possible to derive an equivalent and tractable surrogate objective known as the *variational*

[4] In fact, $\mathrm{KL}\left(q_\phi(\mathbf{z}) \parallel p_\theta(\mathbf{z}|\mathbf{y})\right) = \infty$ if $\exists \mathbf{z}\left(q_\phi(\mathbf{z}) > 0 \wedge p_\theta(\mathbf{z}|\mathbf{y}) = 0\right)$.

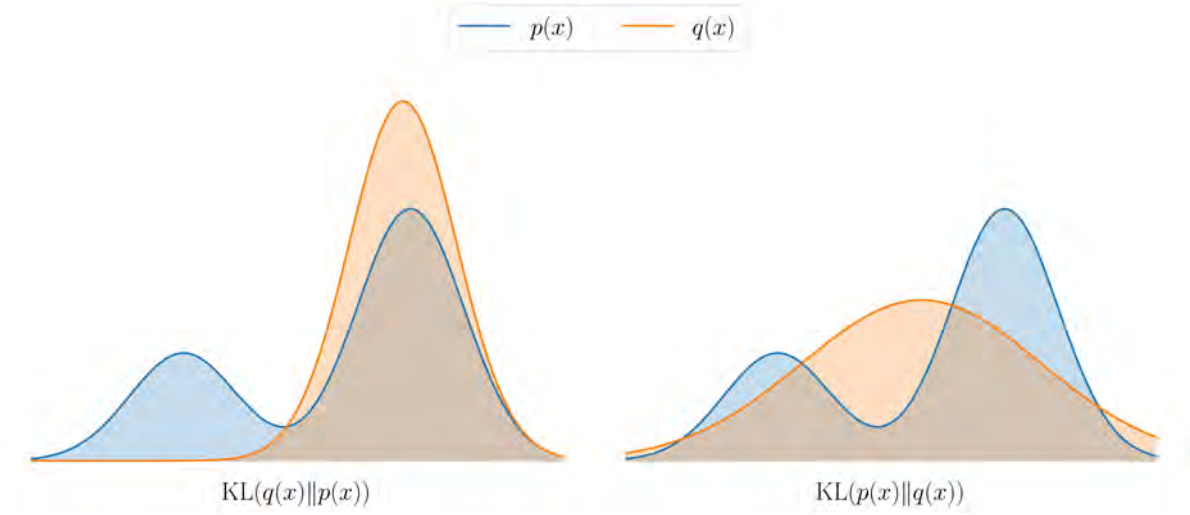


Fig. 2.1 A cartoon illustration of the approximate posteriors which minimise the exclusive KL divergence (left) and inclusive KL divergence (right).

free-energy, $\mathcal{F}_{\text{VFE}}$:

$$\begin{aligned}
\text{KL}\left(q_\phi(\mathbf{z}) \parallel p_\theta(\mathbf{z}|\mathbf{y})\right) &= \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{y})}\right] \\
&= \underbrace{\mathbb{E}_{q_\phi(\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{z})}{p_\theta(\mathbf{y},\mathbf{z})}\right]}_{\mathcal{F}_{\text{VFE}}} + \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log p(\mathbf{y})\right] \\
&= \mathcal{F}_{\text{VFE}} + \log p_\theta(\mathbf{y}).
\end{aligned} \tag{2.9}$$

Since the log marginal likelihood is constant with respect to ϕ, the variational objective is equivalent to minimising $\mathcal{F}_{\text{VFE}}$:

$$\begin{aligned}
\phi^* &= \arg\min_\phi \ \mathcal{F}_{\text{VFE}} \\
&= \arg\min_\phi \ \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{z})}{p_\theta(\mathbf{y},\mathbf{z})}\right]
\end{aligned} \tag{2.10}$$

It is more common to refer to the negative of the variational free energy, known as the *variational lower bound* or *evidence lower bound* (ELBO):

$$\mathcal{L}_{\text{ELBO}} \coloneqq -\mathcal{F}_{\text{VFE}} = \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{y},\mathbf{z})}{q_\phi(\mathbf{z})}\right], \tag{2.11}$$

which bounds the log marginal likelihood[5] below by an amount equal to $\text{KL}\left(q_\phi(\mathbf{z}) \parallel p_\theta(\mathbf{z}|\mathbf{y})\right)$:

$$\mathcal{L}_{\text{ELBO}} = \log p_\theta(\mathbf{y}) - \text{KL}\left(q_\phi(\mathbf{z}) \parallel p_\theta(\mathbf{z}|\mathbf{y})\right). \tag{2.12}$$

[5] Also known as the *evidence*.

The tightness of the bound improves as $q_\phi(\mathbf{z})$ more closely approximates $p_\theta(\mathbf{z}|\mathbf{y})$. Importantly, VI can be interpreted as jointly approximating the two objects of primary interest in probabilistic machine learning: the log marginal likelihood, $\mathcal{L}_{\text{ELBO}} \approx p_\theta(\mathbf{y})$, and the posterior, $q_\phi(\mathbf{z}) \approx p_\theta(\mathbf{z}|\mathbf{y})$. Due to the relationship between the ELBO and log marginal likelihood, the ELBO can be jointly optimised with respect to both the model and variational parameters[6]. Crucially, VI unifies the task of approximating learning and inference into a single optimisation objective:

$$\theta^*, \phi^* = \arg\max_{\theta,\phi} \ \mathcal{L}_{\text{ELBO}}. \tag{2.13}$$

This duality makes VI a particularly attractive framework for machine learning practitioners.

2.2.2 Monte Carlo Variational Inference

The optimisation objective of (2.13) has no closed form solution in general. Furthermore, for many models of interest the ELBO cannot be evaluated analytically[7]. An approach that is widely adopted in practice is Monte Carlo VI (MC-VI), or equivalently black box VI (Ranganath et al., 2014), which considers a Monte Carlo approximation to the ELBO:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{y},\mathbf{z})}{q_\phi(\mathbf{z})}\right] \approx \frac{1}{S}\sum_{s=1}^{S} \log p(\mathbf{z}^s|\mathbf{y}) - \log q_\phi(\mathbf{z}^s) + \log p_{\theta_1}(\mathbf{z}^s) \tag{2.14}$$

where $\mathbf{z}^s \sim q_\phi(\mathbf{z})$. Although evaluating the ELBO provides a useful basis for determining the fit of the model to the data, we are much more interested in the gradients of $\mathcal{L}_{\text{ELBO}}$ necessary for gradient based optimisation. We resort to the use of stochastic gradient ascent, which follows the path of 'noisy' Monte Carlo approximations to the gradient.

2.2.3 Stochastic Optimisation of the Variational Objective

A Monte Carlo estimate of the gradient with respect to θ can be obtained by taking the derivative of (2.14). However, naïvely taking the derivatives of (2.14) with respect to ϕ neglects the dependency of the sampling procedure on the variational parameters. This is because the derivative operator cannot merely be moved inside the expectation:

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{y},\mathbf{z})}{q_\phi(\mathbf{z})}\right] \neq \mathbb{E}_{q_\phi(\mathbf{z})}\left[\nabla_\phi \log \frac{p_\theta(\mathbf{y},\mathbf{z})}{q_\phi(\mathbf{z})}\right]. \tag{2.15}$$

In this section, we discuss two methods for accounting for this dependency.

[6] A word of caution: it is possible that the ELBO is not uniformly tight, and its maximum is biased away from the maximum of the log marginal likelihood (Turner and Sahani, 2011).

[7] In DLVMs, as is often the case, this is due to the presence of non-linearities in the likelihood: evaluating $\mathbb{E}_{q_\phi(\mathbf{z})}\left[\log p_{\theta_2}(\mathbf{y}|\mathbf{z})\right]$ requires the propagation of $q_\phi(\mathbf{z})$ through these non-linearities which, for even simple $q_\phi(\mathbf{z})$, is analytically intractable.

Score Function Estimator

The first method relies on the so called log-derivative trick, which uses the identity

$$\nabla p(\mathbf{x}) = p(\mathbf{x}) \nabla \log p(\mathbf{x}). \tag{2.16}$$

Application of the log-derivative trick to the ELBO proceeds as follows:

$$\begin{aligned}
\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})} \left[\log \frac{p_\theta(\mathbf{y}, \mathbf{z})}{q_\phi(\mathbf{z})} \right] &= \nabla_\phi \int q_\phi(\mathbf{z}) \log \frac{p_\theta(\mathbf{y}, \mathbf{z})}{q_\phi(\mathbf{z})} d\mathbf{z} \\
&= \int \nabla_\phi \left(q_\phi(\mathbf{z}) \log \frac{p_\theta(\mathbf{y}, \mathbf{z})}{q_\phi(\mathbf{z})} \right) d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}) \nabla_\phi \left(\log q_\phi(\mathbf{z}) \right) \log \frac{p_\theta(\mathbf{y}, \mathbf{z})}{q_\phi(\mathbf{z})} d\mathbf{z} + \int q_\phi(\mathbf{z}) \nabla_\phi \left(\log \frac{p_\theta(\mathbf{y}, \mathbf{z})}{q_\phi(\mathbf{z})} \right) d\mathbf{z} \\
&= \mathbb{E}_{q_\phi(\mathbf{z})} \left[\log \frac{p_\theta(\mathbf{y}, \mathbf{z})}{q_\phi(\mathbf{z})} \nabla_\phi \log q_\phi(\mathbf{z}) \right] - \underbrace{\mathbb{E}_{q_\phi(\mathbf{z})} \left[\nabla_\phi \log q_\phi(\mathbf{z}) \right]}_{=0} \\
&= \mathbb{E}_{q_\phi(\mathbf{z})} \left[\log \frac{p_\theta(\mathbf{y}, \mathbf{z})}{q_\phi(\mathbf{z})} \nabla_\phi \log q_\phi(\mathbf{z}) \right],
\end{aligned} \tag{2.17}$$

where we have used Leibniz's rule for differentiation under the integral sign[8] and the fact that

$$\int q_\phi(\mathbf{z}) \nabla_\phi \log q_\phi(\mathbf{z}) d\mathbf{z} = \int \frac{q_\phi(\mathbf{z})}{q_\phi(\mathbf{z})} \nabla_\phi q_\phi(\mathbf{z}) d\mathbf{z} = \nabla_\phi \underbrace{\int q_\phi(\mathbf{z}) d\mathbf{z}}_{1} = 0. \tag{2.18}$$

With the derivative operator now inside the expectation, an unbiased Monte Carlo estimate can be constructed. We shall refer to the resulting estimator as the *score function estimator*, also known as REINFORCE[9] (Sutton et al., 2000):

$$\nabla_\phi \mathcal{L}_{\text{ELBO}} \approx \frac{1}{S} \sum_{s=1}^{S} \log \frac{p_\theta(\mathbf{y}, \mathbf{z}^s)}{q_\phi(\mathbf{z}^s)} \nabla_\phi \log q_\phi(\mathbf{z}^s) \tag{2.19}$$

where $\mathbf{z}^s \sim q_\phi(\mathbf{z})$. In practice, without the implementation of variance reduction techniques[10], the variance of the score function estimator is large. This cripples its effectiveness in optimising the variational objective (Mohamed et al., 2019; Paisley et al., 2012; Ranganath et al., 2014).

[8] Specifically, $\lim_{a \to -\infty, b \to \infty} \frac{\partial}{\partial x} \int_a^b f(x, z) dz = \lim_{a \to -\infty, b \to \infty} \int_a^b \frac{\partial}{\partial x} f(x, z) dz$.

[9] The term 'score function' refers to the derivative of a log probability density with respect to its parameters. The name 'REINFORCE' refers to the weighting of point estimates of the gradient, $\nabla_\phi \log q_\phi(\mathbf{z}^s)$, by the quantity $\log \frac{p_\theta(\mathbf{y}, \mathbf{z}^s)}{q_\phi(\mathbf{z}^s)}$. The weighting is very positive when $p_\theta(\mathbf{y}, \mathbf{z}) \gg q_\phi(\mathbf{z})$ - we would like $q_\phi(\mathbf{z})$ to increase here - and very negative when $q_\phi(\mathbf{z}) \gg p_\theta(\mathbf{y}, \mathbf{z})$ - we would like $q_\phi(\mathbf{z})$ to decrease here.

[10] Such as control variables and Rao-Blackwellisation. See Li (2017) for an excellent review.

Path Derivative Estimator

An alternative estimator can be constructed using a change of variables. Provided the random variable $\mathbf{x} \sim p(\mathbf{x})$ can be expressed as a differentiable transformation of another random variable $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$,

$$\mathbf{x} = g(\boldsymbol{\epsilon}), \tag{2.20}$$

then expectations under $p(\mathbf{x})$ can be rewritten as equivalent expectations under $p(\boldsymbol{\epsilon})$:

$$\mathbb{E}_{p(\mathbf{x})}\left[f(\mathbf{x})\right] = \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[f(g(\boldsymbol{\epsilon}))\right]. \tag{2.21}$$

This technique is referred to as the law of unconscious statisticians (LOTUS) (Grimmett et al., 2020), or reparameterisation trick (Kingma and Welling, 2014). Equipped with the reparameterisation trick, the ELBO can be reformulated as

$$\mathbb{E}_{q_\phi(\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{y}, \mathbf{z})}{q_\phi(\mathbf{z})}\right] = \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\log \frac{p_\theta(\mathbf{y}, g(\boldsymbol{\epsilon}, \phi))}{q_\phi(g(\boldsymbol{\epsilon}, \phi))}\right]. \tag{2.22}$$

Since $p(\boldsymbol{\epsilon})$ has no dependence on ϕ, the derivative operator can be taken inside the expectation where the chain rule is applied:

$$\begin{aligned}
\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{y}, \mathbf{z})}{q_\phi(\mathbf{z})}\right] &= \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\nabla_\phi \log \frac{p_\theta(\mathbf{y}, g(\boldsymbol{\epsilon}, \phi))}{q_\phi(g(\boldsymbol{\epsilon}, \phi))}\right] \\
&= \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\nabla_\phi \log \frac{p_\theta(\mathbf{y}, \mathbf{z})}{q_\phi(\mathbf{z})}\right] + \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\nabla_{\mathbf{z}} \log \frac{p_\theta(\mathbf{y}, \mathbf{z})}{q_\phi(\mathbf{z})} \nabla_\phi g(\boldsymbol{\epsilon}, \phi)\right] \\
&= -\underbrace{\mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\nabla_\phi \log q_\phi(\mathbf{z})\right]}_{=0} + \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\nabla_{\mathbf{z}} \log \frac{p_\theta(\mathbf{y}, \mathbf{z})}{q_\phi(\mathbf{z})} \nabla_\phi g(\boldsymbol{\epsilon}, \phi)\right] \\
&= \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\nabla_{\mathbf{z}} \log \frac{p_\theta(\mathbf{y}, \mathbf{z})}{q_\phi(\mathbf{z})} \nabla_\phi g(\boldsymbol{\epsilon}, \phi)\right].
\end{aligned} \tag{2.23}$$

The quantity

$$\nabla_{\mathbf{z}} \log \frac{p_\theta(\mathbf{y}, \mathbf{z})}{q_\phi(\mathbf{z})} \nabla_\phi g(\boldsymbol{\epsilon}, \phi) \tag{2.24}$$

is known as the path derivative - it accounts for the dependence on ϕ through the deterministic transformation $g(\boldsymbol{\epsilon}, \phi)$. The *path derivative estimator* is given by

$$\nabla_\phi \mathcal{L}_{\text{VI}} \approx \frac{1}{S} \sum_{s=1}^{S} \nabla_{\mathbf{z}^s} \log \frac{p_\theta(\mathbf{y}, \mathbf{z}^s)}{q_\phi(\mathbf{z}^s)} \nabla_\phi g(\boldsymbol{\epsilon}^s, \phi). \tag{2.25}$$

where $\boldsymbol{\epsilon}^s \sim p(\boldsymbol{\epsilon})$ and $\mathbf{z}^s = g(\boldsymbol{\epsilon}^s, \phi)$. Whereas the score function estimator only uses the scalar values of $p_\theta(\mathbf{y}, \mathbf{z})$, the path derivative estimator uses its gradient. This renders the path derivative estimator inapt in settings where computation of the gradient is intractable. For

DLVMs, however, the gradient can often be computed with relative ease owing to the efficiency of the backpropagation algorithm (Kelley, 1960). Further, the path derivative estimator has been shown empirically to exhibit much lower variance than the score function estimator (Kucukelbir et al., 2017; Mohamed et al., 2019).

2.3 Variational Autoencoders

A popular choice for the approximate distribution in VI are those that are fully factorised across latent variables, with each factor having its own set of local variational parameters ϕ_i:

$$q_\phi(\mathbf{z}) = \prod_i q_{\phi_i}(z_i). \tag{2.26}$$

This is commonly referred to as the mean-field approximation, whose use in VI is widespread (Blei et al., 2003). The use of fully factorised approximate posteriors is motivated by their efficiency and flexibility. Yet, they are incapable of modelling posterior dependencies between latent variables and risk severely underestimating the posterior uncertainty, especially when applied to time-series (MacKay, 2003; Turner and Sahani, 2011). The severity of these limitations are application specific, so should always be taken into account by machine learning practitioners.

Although mean-field VI has found success in many domains, its application to LVMs with local latent variables is ill-suited as the form of the approximate posterior requires the introduction of a set of variational parameters for each observation. This leads to two major ramifications: first, the number of variational parameters grows linearly with the number of data points which swiftly makes maximising the variational objective computationally prohibitive; second, whenever new observations are made, the corresponding variational parameters must be re-optimised. Thus, inference cannot be performed over test data without additional training.

2.3.1 Amortised Inference

In their seminal work, Kingma and Welling (2014) develop the variational autoencoder (VAE), a family of models that partners DLVMs with efficient inference. The core insight in the development of the VAE was the use of a DNN - known as the *inference network* - to map from observations to local parameters of the approximate posterior distribution, with the weights and biases of the inference network becoming the new variational parameters. For fully factorised Gaussian approximate posteriors, this gives rise to

$$\begin{aligned} q_\phi(\mathbf{z}) &= \prod_{n=1}^{N} \mathcal{N}\left(\mathbf{z}_n; \boldsymbol{\mu}_\phi(\mathbf{y}_n), \operatorname{diag} \boldsymbol{\sigma}_\phi(\mathbf{y}_n)\right) \\ \left(\boldsymbol{\mu}_\phi(\mathbf{y}_n),\ \log \boldsymbol{\sigma}_\phi(\mathbf{y}_n)\right) &= g_\phi(\mathbf{y}_n) \end{aligned} \tag{2.27}$$

where $g_\phi(\cdot)$ represents the inference network. A popular interpretation of VAEs is as probabilistic equivalents of autoencoder models. In turn, the DNN that parameterises the likelihood is often referred to as the *decoder* and the inference network the *encoder*.

The technique of sharing variational parameters across data points is known as amortised inference (Gershman and Goodman, 2014; Kingma and Welling, 2014; Rezende et al., 2014), the advantages of which are twofold:

1. The number of variational parameters remains fixed with respect to the size of the dataset. This implies fewer variational parameters for large datasets, improving the computational efficiency and relaxing the memory requirements.

2. The inference network learns to map from observations to local parameters of the approximate distribution. Thus, the VAE can perform inference on unseen data without the need to make any refinements to the variational parameters. Furthermore, test time inference comes at the small cost of a single pass through the inference network.

Of course, these benefits do not come without cost. Rather than freely optimising each parameter of the approximate posterior, they are now tied together through the inference network. The optimal amortised approximate posterior, $q_{\phi^*_{\text{AM}}}(\mathbf{z})$, is strictly worse than the optimal mean-field approximate posterior, $q_{\phi^*_{\text{MF}}}(\mathbf{z})$, in the sense that

$$\text{KL}\left(q_{\phi^*_{\text{AM}}}(\mathbf{z}) \parallel p_\theta(\mathbf{z}|\mathbf{y})\right) > \text{KL}\left(q_{\phi^*_{\text{MF}}}(\mathbf{z}) \parallel p_\theta(\mathbf{z}|\mathbf{y})\right). \tag{2.28}$$

The difference is known as the amortisation gap (Cremer et al., 2018). By increasing the flexibility of the inference network, the amortisation gap can be made arbitrarily small.

The probabilistic DLVMs specified by VAEs routinely use a standard normal prior over latent variables, such that the probabilistic model becomes

$$\begin{aligned} \mathbf{z} &\sim \prod_{n=1}^{N} \mathcal{N}\left(\mathbf{z}_n; \mathbf{0}, \mathbf{I}\right) \\ \mathbf{y}|\mathbf{z} &\sim \prod_{n=1}^{N} p_\theta(\mathbf{y}_n|\mathbf{z}_n). \end{aligned} \tag{2.29}$$

The use of a standard normal prior offers a number of benefits, the principal one being that both the joint distribution $p_\theta(\mathbf{y}, \mathbf{z})$ and approximate posterior $q_\phi(\mathbf{z})$ become fully factorised across observations. A corollary of this is that the ELBO and its gradients decompose into a sum over individual data points, lending itself to efficient optimisation using mini-batches of data. Notwithstanding the computational benefits, its use completely removes any dependencies between observations - doing so is only valid when the *iid* assumption holds.

2.4 Gaussian Processes

Gaussian processes (GPs) are a family of models that describe a probability distribution over a function. The class of distributions described are such that any finite number of function evaluations are jointly Gaussian distributed (Rasmussen and Williams, 2005). A GP is fully specified by its mean function $m_\theta(\mathbf{x}) = \mathbb{E}\left[f(\mathbf{x})\right]$ and covariance function, or kernel, $k_\theta(\mathbf{x}, \mathbf{x}') = \mathrm{Cov}\left[f(\mathbf{x}), f(\mathbf{x}')\right]$:

$$f \sim \mathcal{GP}\left(m_\theta(\mathbf{x}), k_\theta(\mathbf{x}, \mathbf{x}')\right), \tag{2.30}$$

where $\mathbf{x}$ and $\mathbf{x}'$ specify the locations at which the function f is evaluated and θ denotes the set of model hyper-parameters. Using a zero mean function[11], the joint distribution over a finite collection of function values $\mathbf{f} = (f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_N))$ is given by

$$p(\mathbf{f}) = \mathcal{N}\left(\mathbf{f}; \mathbf{0}, k_\theta\left(\mathbf{X}, \mathbf{X}\right)\right), \tag{2.31}$$

where $\left[k_\theta\left(\mathbf{X}, \mathbf{X}\right)\right]_{ij} = k_\theta(\mathbf{x}_i, \mathbf{x}_j)$. $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^T$ is known as the *design matrix*. The distinction between model parameters and hyper-parameters is subtle, but nonetheless important. Given the parameters of a model, future predictions are independent of the observed data $\mathcal{D}$. The same does not hold for model hyper-parameters. GPs themselves do not have any model parameters - they belong to the family of *Bayesian nonparametric* models. Rather than conditioning on the observed data through a set of parameters, GPs condition on the observed data directly. Nonparametric models offer significant advantages over their parametric counterparts, most notably in their ability to infer the appropriate degree of model complexity from the observed data (Ghahramani, 2013). However, as we shall see, this flexibility often comes at the cost of a burdening computational complexity.

The covariance function lies at the crux of GP modelling. It captures the underlying properties of the function f such as smoothness, amplitude and periodicity. Two of the most widely used covariance functions are the squared exponential (SE) kernel,

$$k_{\mathrm{SE}}(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2l^2}\right) \tag{2.32}$$

whose hyper-parameters are the lengthscale l and the output variance σ^2, and the periodic kernel,

$$k_{\mathrm{Per}}(x, x') = \sigma^2 \exp\left(-\frac{2\sin^2(\pi|x - x'|/p)}{l^2}\right) \tag{2.33}$$

[11] Only rarely is this a significant limitation since it does not restrict the posterior mean function to be zero also.

whose hyper-parameters are the period p, the lengthscale l and the output variance σ^2 (MacKay, 1998). Figure 2.2 illustrates functions drawn at random from GPs defined using each covariance function.

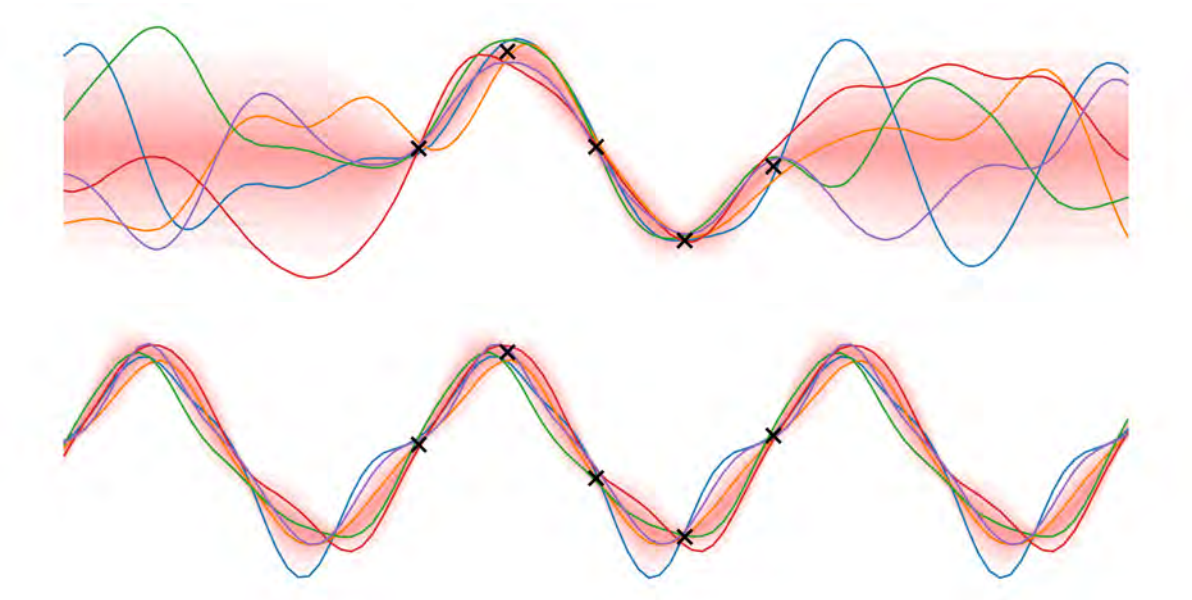


Fig. 2.2 A visualisation of functions drawn from GPs with a SE covariance function (top) and periodic covariance function (bottom). The shades of red correspond to deciles of the posterior predictive distribution at each input location. Coloured lines show samples from the GP. The black crosses show the data being conditioned on.

2.4.1 Gaussian Process Regression

Consider the standard regression task, in which we wish to model a dataset $\mathcal{D}$ consisting of N input and corresponding scalar output pairs, $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$. It is typical to assume the observations y are generated from some latent function $f(\mathbf{x})$ corrupted by additive Gaussian noise:

$$y = f(\mathbf{x}) + \sigma_n \epsilon, \tag{2.34}$$

where $\epsilon \sim \mathcal{N}(\epsilon; 0, 1)$ and σ_n^2 is the noise variance. Placing a GP prior on the latent function gives rise to the probabilistic model

$$\begin{aligned} f &\sim \mathcal{GP}\left(0, k_{\theta_1}(\mathbf{x}, \mathbf{x}')\right) \\ \mathbf{y}|f &\sim \prod_{n=1}^{N} \mathcal{N}\left(y_n; f(\mathbf{x}_n), \sigma_n^2\right), \end{aligned} \tag{2.35}$$

where $\theta = \{\theta_1, \sigma_n^2\}$. Unlike the standard normal prior regularly employed by VAEs, a GP prior explicitly models the dependence between the latent function values corresponding to each observation.

Bayesian inference is primarily concerned with obtaining a posterior distribution over the latent function conditioned on the observations. The posterior distribution is also a Gaussian process:

$$f|\mathbf{y} \sim \mathcal{GP}\left(\hat{m}(\mathbf{x}), \hat{k}(\mathbf{x}, \mathbf{x}')\right) \tag{2.36}$$

with posterior mean and covariance functions given by

$$\begin{aligned} \hat{m}(\mathbf{x}) &= k_{\theta_1}(\mathbf{x}, \mathbf{X})\mathbf{K}_{\mathbf{yy}}^{-1}\mathbf{y} \\ \hat{k}(\mathbf{x}, \mathbf{x}') &= k_{\theta_1}(\mathbf{x}, \mathbf{x}') - k_{\theta_1}(\mathbf{x}, \mathbf{X})\mathbf{K}_{\mathbf{yy}}^{-1}k_{\theta_1}(\mathbf{X}, \mathbf{x}') \end{aligned} \tag{2.37}$$

where $\mathbf{K}_{\mathbf{yy}} = k_{\theta_1}(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I$ is the prior covariance of the observations.

If the properties of the latent function are known a priori, the covariance function and its hyper-parameters can be chosen accordingly to reflect this information. In most settings, however, we are more interested in inferring the properties of the latent function from the observed data. The hyper-parameters that best explain the observed data are those that maximise the log marginal likelihood:

$$\begin{aligned} \theta^* &= \arg\max_{\theta} \ \log p_\theta(\mathbf{y}) \\ &= \arg\max_{\theta} \ -\frac{1}{2}\mathbf{y}^T\mathbf{K}_{\mathbf{yy}}^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_{\mathbf{yy}}| - \frac{N}{2}\log 2\pi. \end{aligned} \tag{2.38}$$

The objective above describes a trade-off between fitting the data ($-\frac{1}{2}\mathbf{y}^T\mathbf{K}_{\mathbf{yy}}^{-1}\mathbf{y}$) and penalising model complexity ($-\frac{1}{2}\log|\mathbf{K}_{\mathbf{yy}}|$), a heuristic known as Occam's razor (MacKay, 1992; Rasmussen and Ghahramani, 2001). Thus, hyper-parameter tuning using the log marginal likelihood is automatically robust to overfitting. This contrasts with parameter tuning through maximisation of the log-likelihood, which does not take into account the generalisability of the parameters[12]. For GPs, the maximum log marginal likelihood forms a reasonable approximation to the *model evidence*

$$p(\mathbf{y}) = \int p_\theta(\mathbf{y})p(\theta)d\theta \approx p_{\theta^*}(\mathbf{y}), \tag{2.39}$$

[12]Recall from Section 2.1 that the parameters of LVMs are also found through maximisation of the log marginal likelihood. Whilst this seems like a contradiction, there is a distinction between the marginal likelihood used in LVMs and that used in GPs. For LVMs, the marginal likelihood refers to integrating out the latent variables from the joint distribution, not the model parameters. For GPs, the marginal likelihood refers to integrating out the model parameters, which has already been achieved as GPs are nonparametric. Thus, learning of model parameters in LVMs through maximisation of the log marginal likelihood does not guard against overfitting, whereas learning of model hyper-parameters in GPs does.

which can in turn be used to perform principled Bayesian model comparison (Rasmussen and Williams, 2005). This is known as the type-II maximum likelihood approximation, valid only when $p_\theta(\mathbf{y})$ is sharply peaked and $p(\theta)$ is flat (MacKay, 1992).

2.4.2 Sparse Gaussian Processes

Evaluating the mean and covariance function of the posterior GP in (2.37) and computing the log marginal likelihood in (2.38) is dominated by the $\mathcal{O}\left(N^3\right)$ cost associated with inverting $\mathbf{K_{yy}}$. For large datasets, this renders exact inference in GPs impracticable. Most research efforts into GPs have been concerned with reducing this computational complexity, and many excellent sparse approximations have been developed. In general, these approximations can be interpreted as performing approximate inference[13] using $M < N$ 'inducing points' $\mathbf{u}$ at 'inducing locations' $\mathbf{Z}$.

One of the pioneering developments in sparse GPs was the use of VI, discussed in detail in Section 2.2. First introduced by Titsias (2009), the approach has been widely adopted as the go-to method for performing approximate inference in GPs. Following Matthews et al. (2016), we introduce a posterior over the latent function f that explicitly parameterises the approximate distribution over inducing points $\mathbf{u}$:

$$q(f) = p(f_{\setminus \mathbf{u}}|\mathbf{u})q(\mathbf{u}) \tag{2.40}$$

where

$$q(\mathbf{u}) = \mathcal{N}\left(\mathbf{u}; \mathbf{m}, \mathbf{S}\right). \tag{2.41}$$

The approximate posterior uses the GP prior $p(f_{\setminus \mathbf{u}}|\mathbf{u})$ in place of the exact posterior $p(f_{\setminus \mathbf{u}}|\mathbf{u}, \mathbf{y})$ and the variational distribution $q(\mathbf{u})$ in place of the posterior $p(\mathbf{u}|\mathbf{y})$. In effect, the latent variables $f_{\setminus \mathbf{u}}$ are affected by the data only through $\mathbf{u}$. The variational parameters consist of the inducing locations $\mathbf{Z}$, mean $\mathbf{m}$ and covariance $\mathbf{S}$. Since these variational parameters are shared across the entire latent function, this framework can be interpreted as an alternative form of amortised inference to that used in VAEs.

Whereas Section 2.2 discussed the minimisation of the KL divergence between two distributions over finite dimensional vectors, here we are interested in minimising the KL divergence between two distributions over infinite dimensional functions. Fortunately, as demonstrated by Matthews

[13] Using the framework of power expectation propagation, Bui (2018) unified two seemingly disparate sparse GP methods under the single hood of approximate inference.

et al. (2016), the formalism is near identical. The ELBO is given by

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q(f)}\left[\log \frac{p(\mathbf{y}|\mathbf{f})\cancel{p(f_{\setminus \mathbf{u}}|\mathbf{u})}p(\mathbf{u})}{\cancel{p(f_{\setminus \mathbf{u}}|\mathbf{u})}q(\mathbf{u})}\right] = \mathbb{E}_{q(\mathbf{f},\mathbf{u})}\left[\log \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{u})}{q(\mathbf{u})}\right]$$
$$= \sum_{n=1}^{N} \mathbb{E}_{q(f(\mathbf{x}_n))}\left[\log p(y_n|f(\mathbf{x}_n))\right] - \text{KL}\left(q(\mathbf{u}) \parallel p(\mathbf{u})\right). \tag{2.42}$$

Whilst an analytical solution to the maximisation of (2.42) can be found with respect to $\mathbf{m}$ and $\mathbf{S}$, the uncollapsed bound lends itself to stochastic optimisation using MC-VI (Hensman et al., 2013). The inducing points can be marginalised out of (2.40) to obtain the approximate posterior GP

$$f \sim \mathcal{GP}\left(m_q(\mathbf{x}), k_q(\mathbf{x}, \mathbf{x}')\right) \tag{2.43}$$

with mean and covariance function given by

$$m_q(\mathbf{x}) = k_{\theta_1}(\mathbf{x}, \mathbf{Z}) k_{\theta_1}(\mathbf{Z}, \mathbf{Z})^{-1} \mathbf{m} \tag{2.44}$$

$$k_q(\mathbf{x}, \mathbf{x}') = k_{\theta_1}(\mathbf{x}, \mathbf{x}') - k_{\theta_1}(\mathbf{x}, \mathbf{Z}) k_{\theta_1}(\mathbf{Z}, \mathbf{Z})^{-1} \left(k_{\theta_1}(\mathbf{Z}, \mathbf{Z}) - \mathbf{S}\right) k_{\theta_1}(\mathbf{Z}, \mathbf{Z})^{-1} k_{\theta_1}(\mathbf{Z}, \mathbf{x}'). \tag{2.45}$$

Crucially, computation of $m_q(\mathbf{x})$ and $k_q(\mathbf{x}, \mathbf{x}')$ sidesteps the cost associated with inverting $\mathbf{K_{yy}}$, reducing the computational complexity to $\mathcal{O}\left(NM^2\right)$. Note that the form of the likelihood $p(y_n|f(\mathbf{x}_n))$ is unspecified. Indeed, the VFE approach to sparse GPs has been extended beyond Gaussian likelihoods to non-linear models (Hensman et al., 2015; Sheth et al., 2015).

2.5 Deep Sets

Recently, the use of machine learning models that operate on sets has garnered a great deal of attention from the research community. Most existing models place strong assumptions on the structure of the data, usually in the form of a fixed dimensional vector. Sets are comparatively much less structured than fixed dimensional vectors. A defining property is permutation invariance - that is, the order of objects in a set has no meaning. Further, sets are free to vary in size. The output of any valid machine learning model operating on sets must be invariant to permutations in the order of a potentially arbitrary number of objects in the set.

Formally, a function f is permutation invariant if

$$f(x_1, x_2, \ldots, x_M) = f(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(M)}) \tag{2.46}$$

for all permutations π. A pivotal innovation in the development of permutation invariant set functions is Deep Sets (Zaheer et al., 2017). Remarkably, Zaheer et al. demonstrate that any

valid permutation invariant set function takes the form

$$f(X) = \rho\left(\sum_{x \in X} h(x)\right), \tag{2.47}$$

also known as the sum decomposition. $X = \{x_1, x_2, \ldots, x_M\}$ represents the set of objects $x_m \in \mathfrak{X}$, and $h : \mathfrak{X} \to Z$ and $\rho : Z \to \mathbb{R}$ are universal function approximations. The well known universal approximation theorems establish that DNNs satisfy this condition[14], motivating their use in Deep Sets (Cybenko, 1989; Hornik et al., 1989). Caveats regarding the form of the latent space, Z, depend on the domain of objects in the set, $\mathfrak{X}$. Notably, Zaheer et al. show that for the uncountable universe $\mathfrak{X} = [0, 1]$ and fixed set size M, Deep Sets using $Z = \mathbb{R}^{M+1}$ can represent any function. Wagstaff et al. (2019) extend the result, proving the case when the set size can vary for $|X| \leq M$.

Deep Sets provide a recipe for constructing permutation invariant set functions, which has important repercussions throughout the field of machine learning. Indeed, the ideas embodied by Deep Sets have been independently developed in a wide variety of contexts, including conditional neural processes (Garnelo et al., 2018), the neural statistician (Edwards and Storkey, 2016) and point cloud modelling (Qi et al., 2017). Within the framework of VI, a motivating application of Deep Sets is to construct approximate posteriors that are invariant to the order in which observations are made. This reflects the permutation invariance of the true posterior, and is particularly attractive in settings in which data is continuously being streamed.

[14] GPs with specific kernels also satisfy this condition (Micchelli et al., 2006).

3 | A Family of Spatio-Temporal Variational Autoencoders

In the previous chapter we discussed the development of GPs and VAEs - two distinct classes of probabilistic models suited to datasets of contrasting characteristics. On the one hand, the effectiveness of VAEs in learning low-dimensional latent representations has facilitated the deployment of DLVMs on large, richly structured high-dimensional datasets. Despite this, VAEs are not directly applicable to datasets which exhibit strong correlations such as those observed over space and time. At the core of this inadequacy is the *iid* assumption of the generative process. On the other hand, GPs are a natural choice for modelling data that exhibits strong correlations. However, the extension of GPs to large, high-dimensional datasets is not immediate. In their exact form, multi-output GPs scale cubically with both the number of data points and the number of dimensions (Álvarez et al., 2012), confining their application to small, low-dimensional datasets.

Spatio-temporal datasets arise naturally from a wealth of domains including environmental, social and earth sciences. They are characterised by the presence of strong dependencies across space and time, often taking the form of multi-dimensional observations. Given the complementary strengths of VAEs and GPs, it is instinctive to wish to combine them to model such datasets. In this chapter, we introduce a new family of models that we dub the GP-VAE. Sections 3.1 and 3.2 describe the probabilistic model and establish a principled framework for performing approximate inference. Section 3.3 builds upon this framework, presenting a general template for handling partially observed data and extending the capability of GP-VAEs accordingly. Finally, in Section 3.4 we present a novel amalgamation of sparse GPs and VAEs, developing the sparse GP-VAE which offers substantial computational advantages over the regular GP-VAE and existing sparse GP models.

3.1 The GP-VAE

3.1.1 The Probabilistic Model

Consider the multi-output regression task in which we wish to model a dataset consisting of N D-dimensional input and corresponding P-dimensional output pairs, $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^N$. Let $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N)$ denote the concatenation of outputs and $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^T$ the design matrix. We model each observation as being generated from a diagonal Gaussian distribution parameterised by passing some latent variable $\mathbf{f}_n \in \mathbb{R}^K$ through a decoder DNN. The elements of $\mathbf{f}_n$ correspond to the evaluation of a K-dimensional latent function $f = (f_1, f_2, \ldots, f_K)$ at input $\mathbf{x}_n$. Placing an independent GP prior on each latent function dimension gives rise to the complete probabilistic model:

$$\begin{aligned} f &\sim \prod_{k=1}^{K} \underbrace{\mathcal{GP}\left(0, k_{\theta_{1,k}}(\mathbf{x}, \mathbf{x}')\right)}_{p_{\theta_1}(f_k)} \\ \mathbf{y}|\mathbf{f} &\sim \prod_{n=1}^{N} \underbrace{\mathcal{N}\left(\mathbf{y}_n; \boldsymbol{\mu}_{\theta_2}(\mathbf{f}_n), \text{diag}\, \boldsymbol{\sigma}^2_{\theta_2}(\mathbf{f}_n)\right)}_{p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n)}. \end{aligned} \tag{3.1}$$

where $\mathbf{f}_n = f(\mathbf{x}_n)$. $\theta_1 = \{\theta_{1,k}\}_{k=1}^K$ denotes the set of GP hyper-parameters and θ_2 the parameters of the decoder. In a slight departure from fully rigorous terminology, we shall refer to the set $\theta = \{\theta_1, \theta_2\}$ as the model parameters of the GP-VAE. Note that the dependence of the GP prior $p_{\theta_1}(f_k)$ and likelihood $p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n)$ on the inputs has been suppressed for notational convenience. In contrast to the DLVMs employed by vanilla VAEs, the probabilistic model described in (3.1) explicitly models dependencies between latent variables through the GP prior. The motive of the latent structure is twofold: not only are we concerned with discovering a simpler representation of the observed data, but we also wish to discover the dependencies between these representations that explain the dependencies between observations.

3.1.2 The Structured Approximate Posterior

An implication of the non-linear likelihood is that the posterior distribution over the latent function f, given the observed dataset $\mathcal{D}$, is generally intractable. Rigidly following the standard VAE framework laid out by Kingma and Welling (2014) would see us approximate the true posterior over the N latent function evaluations $\mathbf{f} = (\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N)$ with a fully factorised Gaussian parameterised by an inference network

$$q(\mathbf{f}) = \prod_{n=1}^{N} \mathcal{N}\left(\mathbf{f}_n; \boldsymbol{\mu}_\phi(\mathbf{y}_n), \text{diag}\, \boldsymbol{\sigma}^2_\phi(\mathbf{y}_n)\right). \tag{3.2}$$

The variational parameters ϕ denote the weights and biases of the inference network. However, such an approximate distribution fails to account for the dependence between observations due to the GP prior. Further, it does not specify the approximate posterior over the latent function at inputs not included in $\mathcal{D}$. The role of the GP prior in the probabilistic model, and thus the true posterior, is indispensable - any approximate posterior that fails to account for its presence is ill-suited.

Analogous to its presence in the true posterior, we can choose to explicitly incorporate the GP prior using the *structured approximate posterior* over the entire latent function:

$$q(f) = p_{\theta_1}(f_{\setminus \mathbf{f}}|\mathbf{f})\frac{1}{\mathcal{Z}_q(\theta,\phi)}p_{\theta_1}(\mathbf{f})l_\phi(\mathbf{f}|\mathbf{y}) = \prod_{k=1}^{K}\underbrace{p_{\theta_1}(f_{k\setminus\mathbf{f}_k}|\mathbf{f}_k)\overbrace{\frac{1}{\mathcal{Z}_{q_k}(\theta,\phi)}p_{\theta_1}(\mathbf{f}_k)l_\phi(\mathbf{f}_k|\mathbf{y})}^{q(\mathbf{f}_k)}}_{q(f_k)} \tag{3.3}$$

where $l_\phi(\mathbf{f}_k|\mathbf{y})$ is a fully-factorised Gaussian distribution parameterised by an inference network:

$$l_\phi(\mathbf{f}_k|\mathbf{y}) = \mathcal{N}\left(\mathbf{f}_k; \boldsymbol{\mu}_{\phi,k}, \boldsymbol{\Sigma}_{\phi,k}\right) = \prod_{n=1}^{N}\underbrace{\mathcal{N}\left(f_{nk}; \mu_{\phi,k}(\mathbf{y}_n), \sigma^2_{\phi,k}(\mathbf{y}_n)\right)}_{l_\phi(f_{nk}|\mathbf{y}_n)}. \tag{3.4}$$

Since everything is Gaussian, $q(f)$ defines a product of independent approximate GP posteriors over each latent function dimension f_k. The normalisation constants $\mathcal{Z}_{q_k}(\theta,\phi)$ ensure that $q(\mathbf{f}_k)$, the approximate posterior distribution over the N evaluations of the k^{th} latent dimension $\mathbf{f}_k = (f_{1k}, f_{2k}, \ldots, f_{Nk})$, integrates to one:

$$\mathcal{Z}_{q_k}(\theta,\phi) = \int p_{\theta_1}(\mathbf{f}_k)\prod_{n=1}^{N} l_\phi(f_{nk}|\mathbf{y}_n)d\mathbf{f}_k. \tag{3.5}$$

The explicit inclusion of the GP prior 'diffuses' the distribution parameterised by the inference network across the entire latent function in a manner that is consistent with the probabilistic model[1]. Indeed, the form of the structured approximate posterior is identical to that of the true posterior

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{normalisation constant}} \quad \Longleftrightarrow \quad q(f) = \frac{p_{\theta_1}(f)l_\phi(\mathbf{f}|\mathbf{y})}{\mathcal{Z}_q}. \tag{3.6}$$

Observe that $q(f)$ is equal to the true posterior in a 'pseudo probabilistic model' with pseudo likelihoods $l_\phi(\mathbf{f}_n|\mathbf{y}_n) = \prod_{k=1}^{K} l_\phi(f_{nk}|\mathbf{y}_n)$. Thus, we see that the quality of the approximate posterior distribution improves when each pseudo likelihood closely approximates the true

[1] Moreover, structured variational approximations tend to provide more reasonable uncertainty estimates than mean-field approximations. However, the model parameters found using a structured approximation can be more severely biased away from the true optimum (Turner and Sahani, 2011).

likelihood, $p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n)$. By restricting $l_\phi(\mathbf{f}_n|\mathbf{y}_n)$ to preserve conjugacy with the GP prior, a tractable approximation to the true posterior is recovered. This bears resemblance to expectation propagation, a relationship that we discuss in detail in Chapter 4. In accordance with this pseudo model interpretation, $l_\phi(\mathbf{f}_n|\mathbf{y}_n)$ shall be referred to as the *approximate likelihood*.

The mean and covariance functions of the approximate GP over the k^{th} latent dimension are trivial to compute using the mathematical convenience of Gaussian distributions, and are given by:

$$\begin{aligned} \hat{m}_k(\mathbf{x}) &= k_{f_k\mathbf{f}_k}\left(\mathbf{K}_{\mathbf{f}_k\mathbf{f}_k} + \boldsymbol{\Sigma}_{\phi,k}\right)^{-1}\boldsymbol{\mu}_{\phi,k} \\ \hat{k}_k(\mathbf{x},\mathbf{x}') &= k_{f_k f_k'} - k_{f_k\mathbf{f}_k}\left(\mathbf{K}_{\mathbf{f}_k\mathbf{f}_k} + \boldsymbol{\Sigma}_{\phi,k}\right)^{-1} k_{\mathbf{f}_k f_k}. \end{aligned} \tag{3.7}$$

We have adopted the shorthand notation $k_{f_k f_k'} = k_{\theta_{1,k}}(\mathbf{x},\mathbf{x}')$ and $\mathbf{K}_{\mathbf{f}_k\mathbf{f}_k} = k_{\theta_{1,k}}(\mathbf{X},\mathbf{X})$. See Appendix A.1 for a complete derivation. In contrast to the fully-factorised approximate posterior in (3.2), the mean and covariance functions can be evaluated at any input location to obtain an approximation to the posterior predictive distribution over the corresponding observation. Further, the posterior predictive distribution accounts for the dependencies between the unknown and observed variables due to the probabilistic model.

We refer to the combination of the aforementioned probabilistic model and structured approximate posterior as the GP-VAE.

3.1.3 The Posterior Predictive Distribution

Given the model $p_\theta(f,\mathbf{y})$, the posterior predictive distribution over observation $\mathbf{y}_*$ is given by

$$p_\theta(\mathbf{y}_*|\mathbf{y}) = \int p_\theta(\mathbf{y}_*|\mathbf{f}_*)p(\mathbf{f}_*|\mathbf{y})d\mathbf{f}_*. \tag{3.8}$$

This can be approximated as

$$\begin{aligned} p_\theta(\mathbf{y}_*|\mathbf{y}) &\approx \int p_\theta(\mathbf{y}_*|\mathbf{f}_*)q(\mathbf{f}_*)d\mathbf{f}_* \\ &= \mathbb{E}_{q(\mathbf{f}_*)}\left[p_\theta(\mathbf{y}_*|\mathbf{f}_*)\right] \end{aligned} \tag{3.9}$$

where we have used the approximate posterior $q(\mathbf{f}_*)$ in place of the true posterior $p(\mathbf{f}_*|\mathbf{y})$. Unfortunately, the integral in (3.9) is intractable. Instead, we can approximate the approximate posterior predictive distribution as a mixture of Gaussians:

$$p_\theta(\mathbf{y}_*|\mathbf{y}) \approx \frac{1}{S}\sum_{s=1}^{S} \underbrace{\mathcal{N}\left(\mathbf{y}_*; \boldsymbol{\mu}_{\theta_2}(\mathbf{f}_*^s), \operatorname{diag}\boldsymbol{\sigma}_{\theta_2}^2(\mathbf{f}_*^s)\right)}_{p_\theta(\mathbf{y}_*|\mathbf{f}_*^s)} \tag{3.10}$$

where $\mathbf{f}_*^s \sim q(\mathbf{f}_*)$.

3.2 Monte Carlo Variational Inference

Learning and inference in the GP-VAE are concerned with determining the model parameters θ and variational parameters ϕ, respectively. We have seen how these objectives can be attained simultaneously by maximising the variational lower bound, or ELBO. Following Chapter 2, the ELBO for the structured approximate posterior is given by

$$\begin{aligned}\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q(f)}\left[\log \frac{p_\theta(\mathbf{y}, f)}{q(f)}\right] &= \mathbb{E}_{q(f)}\left[\log \frac{\cancel{p_{\theta_1}(f_{\setminus \mathbf{f}}|\mathbf{f})}\,\cancel{p_{\theta_1}(\mathbf{f})}\,p_{\theta_2}(\mathbf{y}|\mathbf{f})}{\cancel{p_{\theta_1}(f_{\setminus \mathbf{f}}|\mathbf{f})}\,\frac{1}{\mathcal{Z}_q(\theta,\phi)}\,\cancel{p_{\theta_1}(\mathbf{f})}\,l_\phi(\mathbf{f}|\mathbf{y})}\right] \\ &= \mathbb{E}_{q(\mathbf{f})}\left[\log \frac{p_{\theta_2}(\mathbf{y}|\mathbf{f})}{l_\phi(\mathbf{f}|\mathbf{y})}\right] + \log \mathcal{Z}_q(\theta,\phi). \end{aligned} \tag{3.11}$$

The cancellation of the GP prior ensures that the ELBO can be evaluated by sampling the infinite-dimensional approximate posterior at the finite number of locations included in the observed data. The first term in (3.11) also appears in the ELBO for the standard VAE; however, the final term differs due to the presence of the GP prior.

Neither the ELBO or its gradients are analytically tractable. In this section, we derive the estimators necessary for performing MC-VI.

3.2.1 Estimating the ELBO

The intractability of the ELBO arises due to the non-linear likelihood function. In particular, the quantity $\mathbb{E}_{q(\mathbf{f})}\left[\log p_{\theta_2}(\mathbf{y}|\mathbf{f})\right]$ amounts to propagating a Gaussian distribution, $q(\mathbf{f})$, through the non-linear decoder network. A straightforward but nonetheless effective workaround is to use the Monte Carlo estimate

$$\mathbb{E}_{q(\mathbf{f})}\left[\log p_{\theta_2}(\mathbf{y}|\mathbf{f})\right] \approx \frac{1}{S}\sum_{s=1}^{S}\sum_{n=1}^{N} \log p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n^s) \tag{3.12}$$

where $\mathbf{f}^s \sim q(\mathbf{f})$. The number of samples, S, can be chosen to achieve an arbitrary degree of accuracy. Fortunately, the quantity $\mathbb{E}_{q(\mathbf{f})}\left[\log l_\phi(\mathbf{f}|\mathbf{y})\right]$ involves only the first and second moments

of $q(\mathbf{f})$ and so a closed form solution exists:

$$\begin{aligned}
\mathbb{E}_{q(\mathbf{f})}\left[\log l_\phi(\mathbf{f}|\mathbf{y})\right] &= \sum_{k=1}^{K}\sum_{n=1}^{N} \mathbb{E}_{q(\mathbf{f}_k)}\left[\log l_\phi(f_{nk}|\mathbf{y}_n)\right] \\
&= \sum_{k=1}^{K}\sum_{n=1}^{N} \mathbb{E}_{q(\mathbf{f}_k)}\left[-\frac{(f_{nk}-\mu_{\phi,k}(\mathbf{y}_n))^2}{2\sigma^2_{\phi,k}(\mathbf{y}_n)} - \frac{1}{2}\log|2\pi\sigma^2_{\phi,k}(\mathbf{y}_n)|\right] \\
&= \sum_{k=1}^{K}\sum_{n=1}^{N} -\frac{\left[\hat{\mathbf{\Sigma}}_k\right]_{nn} + (\hat{\mu}_{k,n}-\mu_{\phi,k}(\mathbf{y}_n))^2}{2\sigma^2_{\phi,k}(\mathbf{y}_n)} - \frac{1}{2}\log|2\pi\sigma^2_{\phi,k}(\mathbf{y}_n)| \\
&= \sum_{k=1}^{K}\sum_{n=1}^{N} \log\mathcal{N}\left(\hat{\mu}_{k,n};\mu_{\phi,k}(\mathbf{y}_n),\sigma^2_{\phi,k}(\mathbf{y}_n)\right) - \frac{\left[\hat{\mathbf{\Sigma}}_k\right]_{nn}}{2\sigma^2_{\phi,k}(\mathbf{y}_n)} \\
&= \sum_{k=1}^{K} \log\mathcal{N}\left(\hat{\boldsymbol{\mu}}_k;\boldsymbol{\mu}_{\phi,k},\mathbf{\Sigma}_{\phi,k}\right) - \sum_{n=1}^{N}\frac{\left[\hat{\mathbf{\Sigma}}_k\right]_{nn}}{2\sigma^2_{\phi,k}(\mathbf{y}_n)}.
\end{aligned} \tag{3.13}$$

where $\hat{\boldsymbol{\mu}}_k$ and $\hat{\mathbf{\Sigma}}_k$ are the mean and covariance matrix of $q(\mathbf{f}_k)$, which can be found by evaluating the mean and covariance functions given in (3.7) at inputs $\mathbf{X}$. Similarly, a closed form solution to $\log\mathcal{Z}_q(\theta,\phi)$ exists. First, note that each $\mathcal{Z}_{q_k}(\theta,\phi)$ can be re-written as a convolution between two Gaussians:

$$\begin{aligned}
\mathcal{Z}_q(\theta,\phi) = \prod_{k=1}^{K}\mathcal{Z}_{q_k}(\theta,\phi) &= \prod_{k=1}^{K}\int p_{\theta_1}(\mathbf{f}_k)l_\phi(\mathbf{f}_k|\mathbf{y})d\mathbf{f}_k \\
&= \prod_{k=1}^{K}\int \mathcal{N}\left(\mathbf{f}_k;\mathbf{0},\mathbf{K}_{\mathbf{f}_k\mathbf{f}_k}\right)\mathcal{N}\left(\mathbf{f}_k;\boldsymbol{\mu}_{\phi,k},\mathbf{\Sigma}_{\phi,k}\right)d\mathbf{f}_k \\
&= \prod_{k=1}^{K}\int \mathcal{N}\left(\mathbf{f}_k;\mathbf{0},\mathbf{K}_{\mathbf{f}_k\mathbf{f}_k}\right)\mathcal{N}\left(\boldsymbol{\mu}_{\phi,k}-\mathbf{f}_k;\mathbf{0},\mathbf{\Sigma}_{\phi,k}\right)d\mathbf{f}_k.
\end{aligned} \tag{3.14}$$

The convolution between two Gaussians is also Gaussian, with mean and covariance given by the summation of the means and covariances of the original Gaussians. Thus, $\log\mathcal{Z}_q(\theta,\phi)$ can be concisely expressed as

$$\log\mathcal{Z}_q(\theta,\phi) = \sum_{k=1}^{K}\underbrace{\log\mathcal{N}\left(\boldsymbol{\mu}_{\phi,k};\mathbf{0},\mathbf{K}_{\mathbf{f}_k\mathbf{f}_k}+\mathbf{\Sigma}_{\phi,k}\right)}_{\log\mathcal{Z}_{q_k}(\theta,\phi)}. \tag{3.15}$$

Put together, (3.12), (3.13) and (3.15) provide an unbiased estimate of the ELBO. We refer to this estimator as the *semi-analytic ELBO estimator*.

For the sake of computational efficiency, it is important to use Monte Carlo estimators with the least variance so that we can obtain an accurate approximation with as few samples as possible. Intuition suggests that this is achieved using analytic solutions where possible; however, Roeder

et al. (2017) observe that using a Monte Carlo estimate for $\mathbb{E}_{q(\mathbf{f})}\left[l_\phi(\mathbf{f}|\mathbf{y})\right]$ can sometimes achieve lower variance than using an analytic solution. In particular, when the approximate likelihood is 'good' in the sense that $l_\phi(\mathbf{f}|\mathbf{y}) \approx p_{\theta_2}(\mathbf{y}|\mathbf{f})$, the stochasticity of the Monte Carlo samples cancels:

$$\frac{1}{S}\sum_{s=1}^{S}\log\frac{p_{\theta_2}(\mathbf{y}|\mathbf{f}^s)}{l_\phi(\mathbf{f}^s|\mathbf{y})} \approx \frac{1}{S}\sum_{s=1}^{S}\underbrace{\log\frac{p_{\theta_2}(\mathbf{y}|\mathbf{f}^s)}{p_{\theta_2}(\mathbf{y}|\mathbf{f}^s)}}_{0} = 0. \tag{3.16}$$

Although equality is never satisfied using a Gaussian approximation, the resultant estimator may still exhibit lower variance than the semi-analytic ELBO estimator and is commonly used in practice. We shall refer to the estimator as the *doubly-stochastic ELBO estimator*.

3.2.2 Estimating Gradients of the ELBO

Naïvely taking the derivatives of either of the two ELBO estimators neglects not only the dependence of the sampling procedure on variational parameters ϕ, but also its dependence on model parameters θ. Incongruous with the standard VAE, this is a direct implication of including the GP prior in the approximate posterior. Fortunately, we can apply the two techniques discussed in Chapter 2 to form unbiased Monte Carlo estimates. The first uses the log-derivative trick, whereas the second employs LOTUS, or reparameterisation trick (Grimmett et al., 2020; Kingma and Welling, 2014). We refer to these two flavours of estimators as the *score function estimator* and the *path derivative estimator*, respectively. Throughout this section, the operator $\nabla_{(\cdot)}$ shall denote a generalisation of the operators ∇_θ and ∇_ϕ.

Note that these techniques are only necessary when samples are drawn from the approximate posterior. The gradients of the analytic solutions for $\log \mathcal{Z}_q(\theta, \phi)$ and $\mathbb{E}_{q(\mathbf{f})}\left[l_\phi(\mathbf{f}|\mathbf{y})\right]$ are computed with ease by automatic differentiation packages bundled into popular frameworks such as PyTorch (Paszke et al., 2017) and TensorFlow (Abadi et al., 2016), so require no additional approximations. However, akin to the motivation for the doubly-stochastic ELBO estimator, it can sometimes be beneficial to introduce approximations if the additional stochasticity cancels.

Score Function Estimator

The application of the log-derivative trick to $\nabla_{(\cdot)}\mathbb{E}_{q(\mathbf{f})}\left[p_{\theta_2}(\mathbf{y}|\mathbf{f})\right]$ is straightforward and proceeds as follows:

$$\begin{aligned}
\nabla_{(\cdot)}\mathbb{E}_{q(\mathbf{f})}\left[p_{\theta_2}(\mathbf{y}|\mathbf{f})\right] &= \sum_{n=1}^{N} \nabla_{(\cdot)}\mathbb{E}_{q(\mathbf{f}_n)}\left[p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n)\right] \\
&= \sum_{n=1}^{N} \nabla_{(\cdot)} \int q(\mathbf{f}_n) \log p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n) d\mathbf{f}_n \\
&= \sum_{n=1}^{N} \int \nabla_{(\cdot)} \left(q(\mathbf{f}_n) \log p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n)\right) d\mathbf{f}_n \\
&= \sum_{n=1}^{N} \Bigg\{ \int q(\mathbf{f}_n) \nabla_{(\cdot)} \left(\log q(\mathbf{f}_n)\right) \log p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n) d\mathbf{f}_n \\
&\qquad\qquad - \int q(\mathbf{f}_n) \nabla_{(\cdot)} \left(\log p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n)\right) d\mathbf{f}_n \Bigg\} \\
&= \sum_{n=1}^{N} \left\{ \mathbb{E}_{q(\mathbf{f}_n)}\left[\log p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n) \nabla_{(\cdot)} \log q(\mathbf{f}_n)\right] + \mathbb{E}_{q(\mathbf{f}_n)}\left[\nabla_{(\cdot)} \log p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n)\right] \right\}.
\end{aligned} \tag{3.17}$$

With the derivative operator now inside the expectation, an unbiased Monte Carlo estimate can be made. We refer to this estimator used in conjunction with the gradients of the analytic solutions for $\log \mathcal{Z}_q(\theta, \phi)$ and $\mathbb{E}_{q(\mathbf{f})}\left[l_\phi(\mathbf{f}|\mathbf{y})\right]$ as the *semi-analytic score function estimator*. Note that expectations are over $q(\mathbf{f}_n)$, rather than $q(\mathbf{f})$. Sampling $\mathbf{f}_n \sim q(\mathbf{f}_n)$ circumvents the $\mathcal{O}\left(N^3\right)$ computation of the Cholesky decomposition of the K covariance matrices $\{\hat{\mathbf{\Sigma}}_k\}_{k=1}^K$, requiring only their diagonal elements instead. Unfortunately, $\log \mathcal{Z}_q(\theta, \phi)$ still demands computation of their inverses.

Alternatively, the log-derivative trick can also be applied to approximate $\nabla_{(\cdot)}\mathbb{E}_{q(\mathbf{f})}\left[l_\phi(\mathbf{f}|\mathbf{y})\right]$. Starting from (3.11), we have:

$$\begin{aligned}
\nabla_{(\cdot)}\mathcal{L}_{\text{ELBO}} &= \nabla_{(\cdot)}\mathbb{E}_{q(\mathbf{f})}\left[\log \frac{p_{\theta_2}(\mathbf{y}|\mathbf{f})}{l_\phi(\mathbf{f}|\mathbf{y})}\right] + \nabla_{(\cdot)} \log \mathcal{Z}_q(\theta, \phi) \\
&= \nabla_{(\cdot)} \sum_{n=1}^{N} \mathbb{E}_{q(\mathbf{f}_n)}\left[\log \frac{p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n)}{l_\phi(\mathbf{f}_n|\mathbf{y}_n)}\right] + \nabla_{(\cdot)} \log \mathcal{Z}_q(\theta, \phi) \\
&= \sum_{n=1}^{N} \left\{ \mathbb{E}_{q(\mathbf{f}_n)}\left[\log \frac{p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n)}{l_\phi(\mathbf{f}_n|\mathbf{y}_n)} \nabla_{(\cdot)} \log q(\mathbf{f}_n)\right] + \mathbb{E}_{q(\mathbf{f}_n)}\left[\nabla_{(\cdot)} \log \frac{p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n)}{l_\phi(\mathbf{f}_n|\mathbf{y}_n)}\right] \right\} \\
&\quad + \nabla_{(\cdot)} \log \mathcal{Z}_q(\theta, \phi).
\end{aligned} \tag{3.18}$$

When $l_\phi(\mathbf{f}_n|\mathbf{y}_n) \approx p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n)$, the stochasticity of a Monte Carlo estimate for the first term in the summation cancels. However, the same is not true for the second term as the gradi-

ents do not cancel in general. Thus, we can expect a reduction in variance by pulling out $\mathbb{E}_{q(\mathbf{f})}\left[\nabla_{(\cdot)} \log l_\phi(\mathbf{f}|\mathbf{y})\right]$ and using the analytic solution:

$$\begin{aligned}\mathbb{E}_{q(\mathbf{f})}\left[\nabla_{(\cdot)} \log l_\phi(\mathbf{f}|\mathbf{y})\right] = \sum_{n=1}^{N}\sum_{k=1}^{K}\Big[&-\nabla_{(\cdot)} \log \sigma_{\phi,k}(\mathbf{y}_n) \\ &- \left(\left[\hat{\boldsymbol{\Sigma}}_k\right]_{nn} + (\hat{\mu}_{k,n} - \mu_{\phi,k}(\mathbf{y}_n))^2\right)\nabla_{(\cdot)}\left(\frac{1}{2\sigma^2_{\phi,k}(\mathbf{y}_n)}\right) \\ &+ \frac{1}{2\sigma^2_{\phi,k}(\mathbf{y}_n)}(\hat{\mu}_{k,n} - \mu_{\phi,k}(\mathbf{y}_n))\nabla_{(\cdot)}\mu_{\phi,k}(\mathbf{y}_n)\Big].\end{aligned} \tag{3.19}$$

See Appendix A.2 for a complete derivation. Similarly, a Monte Carlo estimate for the derivative $\nabla_{(\cdot)} \log \mathcal{Z}_q(\theta, \phi)$ is not expected to reduce the variance of the estimator and so analytic solutions are preferable. Collectively, we refer to this estimator as the *doubly-stochastic score function estimator*.

Path Derivative Estimator

LOTUS can be applied using the reparameterisation

$$f_{nk} = g_{\theta,\phi}(\epsilon_{nk}) = \hat{\mu}_{k,n} + \left[\hat{\boldsymbol{\Sigma}}_k\right]^{\frac{1}{2}}_{nn}\epsilon_{nk} \tag{3.20}$$

where $\epsilon_{nk} \sim \mathcal{N}(\epsilon_{nk}; 0, 1)$. This allows the expected log-likelihood to be re-written as

$$\mathbb{E}_{q(\mathbf{f})}\left[\log p_{\theta_2}(\mathbf{y}|\mathbf{f})\right] = \sum_{n=1}^{N}\mathbb{E}_{p(\boldsymbol{\epsilon}_n)}\left[\log p_{\theta_2}(\mathbf{y}_n|g_{\phi,\theta}(\boldsymbol{\epsilon}_n))\right] \tag{3.21}$$

where $\boldsymbol{\epsilon}_n = (\epsilon_{n1}, \epsilon_{n2}, \ldots, \epsilon_{nk})$. Since $p(\boldsymbol{\epsilon}_n)$ is independent of θ and ϕ, the derivative operator can be moved inside the expectation, yielding

$$\begin{aligned}\nabla_{(\cdot)}\mathbb{E}_{q(\mathbf{f})}\left[\log p_{\theta_2}(\mathbf{y}|\mathbf{f})\right] &= \sum_{n=1}^{N}\mathbb{E}_{p(\boldsymbol{\epsilon}_n)}\left[\nabla_{(\cdot)} \log p_{\theta_2}(\mathbf{y}_n|g_{\phi,\theta}(\boldsymbol{\epsilon}_n))\right] \\ &= \sum_{n=1}^{N}\Big\{\mathbb{E}_{p(\boldsymbol{\epsilon}_n)}\Big[\underbrace{\nabla_{(\cdot)} \log p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n)}_{\text{score function}}\Big] + \mathbb{E}_{p(\boldsymbol{\epsilon}_n)}\Big[\underbrace{\nabla_{\mathbf{f}_n} \log p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n)\nabla_{(\cdot)}g_{\theta,\phi}(\boldsymbol{\epsilon}_n)}_{\text{path derivative}}\Big]\Big\}.\end{aligned} \tag{3.22}$$

The expression above is composed of a score function term and a path derivative term, both of which can be estimated by drawing samples $\epsilon^s_n \sim p(\boldsymbol{\epsilon}_n)$ and applying the transformation $f_{nk} = g_{\theta,\phi}(\boldsymbol{\epsilon}^s_{nk})$. Used in combination with the gradients of the analytic solutions for $\log \mathcal{Z}_q(\theta, \phi)$ and $\mathbb{E}_{q(\mathbf{f})}\left[\log l_\phi(\mathbf{f}|\mathbf{y})\right]$, we refer to the estimator as the *semi-analytic path derivative estimator*.

As with the log-derivative trick, an alternative estimator is derived by applying LOTUS to approximate $\nabla_{(\cdot)}\mathbb{E}_{q(\mathbf{f})}\left[l_\phi(\mathbf{f}|\mathbf{y})\right]$. Starting from (3.11), we have

$$\begin{aligned}\nabla_{(\cdot)}\mathcal{L}_{\text{ELBO}} &= \nabla_{(\cdot)}\mathbb{E}_{q(\mathbf{f})}\left[\log\frac{p_{\theta_2}(\mathbf{y}|\mathbf{f})}{l_\phi(\mathbf{f}|\mathbf{y})}\right] + \nabla_{(\cdot)}\log\mathcal{Z}_q(\theta,\phi) \\ &= \sum_{n=1}^{N}\Bigg\{\mathbb{E}_{p(\boldsymbol{\epsilon}_n)}\Bigg[\underbrace{\nabla_{(\cdot)}\log\frac{p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n)}{l_\phi(\mathbf{f}_n|\mathbf{y}_n)}}_{\text{score function}}\Bigg] + \mathbb{E}_{p(\boldsymbol{\epsilon}_n)}\Bigg[\underbrace{\nabla_{\mathbf{f}_n}\log\frac{p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n)}{l_\phi(\mathbf{f}_n|\mathbf{y}_n)}\nabla_{(\cdot)}g_{\theta,\phi}(\boldsymbol{\epsilon}_n)}_{\text{path derivative}}\Bigg]\Bigg\} \\ &\quad + \nabla_{(\cdot)}\log\mathcal{Z}_q(\theta,\phi). \end{aligned} \tag{3.23}$$

Whereas the stochasticity of the path derivative term cancels when the approximate posterior is close to the true posterior, the stochasticity of the score function term does not. Similar to the doubly-stochastic score function estimator, we define the *doubly-stochastic path derivative estimator* using analytical solutions for $\mathbb{E}_{q(\mathbf{f})}\left[\nabla_{(\cdot)}\log l_\phi(\mathbf{f}|\mathbf{y})\right]$ and $\log\mathcal{Z}_q(\theta,\phi)$.

As discussed in Chapter 2, the path derivative estimator is reported to have significantly lower variance than the score function estimator when training VAEs, so is preferable (Kingma and Welling, 2014; Paisley et al., 2012). Choosing between the semi-analytic path derivative estimator and the doubly-stochastic path derivative estimator is not so obvious. Whilst the doubly-stochastic path derivative estimator is low-variance when $l_\phi(\mathbf{f}|\mathbf{y}) \approx p_{\theta_2}(\mathbf{y}|\mathbf{f})$, equality can never be satisfied using a Gaussian approximation. This ambiguity motivates the empirical evaluation of the different gradient estimators, which we detail in Chapter 5.

3.2.3 Mini-Batched Stochastic Gradient Ascent

The computational complexity of performing VI in the GP-VAE is dominated by the $\mathcal{O}\left(KN^3\right)$ cost associated with inverting the set of K $N \times N$ matrices, $\{\mathbf{K}_{\mathbf{f}_k\mathbf{f}_k} + \boldsymbol{\Sigma}_{\phi,k}\}_{k=1}^K$. Although this represents a significant saving relative to the $\mathcal{O}\left(N^3P^3\right)$ cost using exact multi-output GPs, it can quickly become burdensome for even moderately sized datasets. Unfortunately, unlike in standard VAEs, the presence of the GP prior includes full covariance matrices which do not permit a decomposition into a sum over individual data points. A pragmatic workaround is to use a biased estimate of the ELBO using $\tilde{N} < N$ data points:

$$\tilde{\mathcal{L}}_{\text{ELBO}}^{\tilde{N}} = \frac{N}{\tilde{N}}\left[\mathbb{E}_{q(\tilde{\mathbf{f}})}\left[\log\frac{p_{\theta_2}(\tilde{\mathbf{y}}|\tilde{\mathbf{f}})}{l_\phi(\tilde{\mathbf{f}}|\tilde{\mathbf{y}})}\right] + \log\tilde{\mathcal{Z}}_q(\theta,\phi)\right]. \tag{3.24}$$

$\tilde{\mathbf{y}}$ and $\tilde{\mathbf{f}}$ denote the mini-batch of $\tilde{N}$ observations and their corresponding latent variables, respectively. The bias is introduced due to the normalisation constant

$$\log\tilde{\mathcal{Z}}_q(\theta,\phi) = \sum_{k=1}^{K}\log\int p_{\theta_1}(\tilde{\mathbf{f}}_k)l_\phi(\tilde{\mathbf{f}}_k|\mathbf{y})d\tilde{\mathbf{f}}_k \tag{3.25}$$

which does not satisfy $\frac{N}{\tilde{N}}\mathbb{E}\left[\log \tilde{\mathcal{Z}}_q(\theta,\phi)\right] = \mathbb{E}\left[\log \mathcal{Z}_q(\theta,\phi)\right]$. Nevertheless, the mini-batch estimator will be a reasonable approximation to the full estimator provided the lengthscale of the GP prior is not too large[2]. In each step of the optimisation of the ELBO, the use of the mini-batch estimator reduces the computational complexity from $\mathcal{O}\left(KN^3\right)$ to $\mathcal{O}\left(K\tilde{N}^3\right)$. For $\tilde{N} \ll N$, this represents a dramatic improvement.

3.3 Partially Observed Data

Partially observed data is a regularly encountered occurrence in spatio-temporal datasets. In principle, the presence of partially observed data has no effect on the Bayesian paradigm of conditioning the posterior distributions on all that is observed. However, for models using inference networks, such as VAEs, this necessitates modifications to the existing architecture. Standard inference networks condition on data through a non-linear transformation of observations to parameters of an approximate posterior. When only part of the observation is available, it is not immediately obvious how this mapping should be achieved. In this section, we introduce and discuss four distinct methods for handling partial observations: zero imputation, PointNet, IndexNet and FactorNet.

3.3.1 The Partial Observation Framework

Formally, let each partial observation $\mathbf{y}_n$ contain a set of observed values $\mathbf{y}_n^o$ and a set of unobserved values $\mathbf{y}_n^u$:

$$\mathbf{y}_n = \mathbf{y}_n^o \cup \mathbf{y}_n^u. \tag{3.26}$$

Let $\mathcal{O}_n$ denote the index set of observed values $\mathbf{y}_n^o$. Assuming observed values are conditionally independent given the latent variables $\mathbf{f}$, the likelihood of the observed data is given by

$$p_{\theta_2}(\mathbf{y}|\mathbf{f}) = \prod_{n=1}^{N} \prod_{p\in\mathcal{O}_n} p_{\theta_2}(y_{np}|\mathbf{f}_n). \tag{3.27}$$

The structured approximate posterior given in (3.7) must be modified such that the approximate likelihood is conditioned only on the observed values $\mathbf{y}_n^o$:

$$q(f) = p_{\theta_1}(f_{\setminus\mathbf{f}}|\mathbf{f}) \underbrace{\frac{1}{\mathcal{Z}_q(\theta,\phi)} p_{\theta_1}(\mathbf{f}) \prod_{n=1}^{N} l_\phi(\mathbf{f}_n|\mathbf{y}_n^o)}_{q(\mathbf{f})}. \tag{3.28}$$

To exploit the benefits of amortised inference, $l_\phi(\mathbf{f}_n|\mathbf{y}_n^o)$ is parameterised by a *partial inference network* with parameters ϕ. The partial inference network maps from the partially observed

[2] In which case the off-diagonal terms in the covariance matrix will be large making the approximation $p_{\theta_1}(\mathbf{f}) = \prod p_{\theta_1}(\tilde{\mathbf{f}})$ extremely crude.

values $\mathbf{y}_n^o$ to the parameters of $l_\phi(\mathbf{f}_n|\mathbf{y}_n^o)$, which is taken to be Gaussian:

$$l_\phi(\mathbf{f}_n|\mathbf{y}_n^o) = \mathcal{N}\left(\mathbf{f}_n; \boldsymbol{\mu}_\phi(\mathbf{y}_n^o), \text{diag}\, \boldsymbol{\sigma}_\phi^2(\mathbf{y}_n^o)\right). \tag{3.29}$$

The partial inference network must be flexible enough to handle any possible permutation of partially observed data. Using the structured approximate posterior given in (3.28), the modified ELBO for the GP-VAE is given by

$$\mathcal{L}_{\text{ELBO}}^o = \sum_{n=1}^{N} \mathbb{E}_{q(\mathbf{f}_n)}\left[\frac{1}{\alpha}\log p_{\theta_2}(\mathbf{y}_n^o|\mathbf{f}_n) - \log l_\phi(\mathbf{f}_n|\mathbf{y}_n^o)\right] + \log \mathcal{Z}_q(\theta, \phi). \tag{3.30}$$

Setting α equal to the proportion of missing data values rescales the partial likelihood to account for variable rates of missingness. To see this, note that an equivalent expression for $p_{\theta_2}(\mathbf{y}_n^o|\mathbf{f}_n)$ is

$$p_{\theta_2}(\mathbf{y}_n^o|\mathbf{f}_n) = \prod_{p\in\mathcal{O}_n} p_{\theta_2}(y_{np}|\mathbf{f}_n) \times \prod_{p'\notin\mathcal{O}_n} 1 \tag{3.31}$$

i.e. equivalent to the fully observed likelihood with $p_{\theta_2}(y_{np'}|\mathbf{f}_n) = 1,\ \forall\ p' \notin \mathcal{O}_n$. As the rate of missingness increases, this biases $p_{\theta_2}(\mathbf{y}_n^o|\mathbf{f}_n)$ towards 1. The scaling factor α acts to counter this effect.

3.3.2 Partial Inference Network

Arguably the most widely used approach for handling partially observed data in VAEs is to impute the missing values with zeros. This has shown to be effective in a variety of settings, including VAE modelling of heterogeneous data (Nazabal et al., 2020) and previous GP-VAE endeavours (Fortuin et al., 2020). Zero imputation is an attractive choice for practitioners as it requires no modifications to the standard inference network: we simply replace the observations $\mathbf{y}_n$ with zero imputed modified observations $\tilde{\mathbf{y}}_n$. Furthermore, since the inference network applies a non-linear transformation to a weighted sum of inputs, the zero imputed data has no effect on the output of the inference network, and in turn the ELBO and its gradients. A major shortcoming of zero imputation is that the inference network can no longer distinguish between a missing value and a true zero. Provided the data is normalised, a true zero corresponds to the mean value across observations. A value being equal to the mean can often be highly informative, especially when the observations are multi-modal and the mean value is in a region of low density.

Instead, we turn towards ideas of Deep Sets (Zaheer et al., 2017), introduced in Chapter 2. Each partial observation can be reinterpreted as a permutation invariant set by coupling the observed value with the dimension index. The advantage of this formulation is that sets are permutation invariant and can vary in size, unlike fixed-dimensional vectors. Thus, an appropriate set function can handle any possible permutation of partially observed data in a principled manner.

Following Zaheer et al. (2017), we define a family of permutation invariant partial inference networks as

$$g_\phi(\mathbf{y}_n^o) \coloneqq \rho_{\phi_2}\left(\sum_{p\in\mathcal{O}_n} h_{\phi_1}(\mathbf{s}_{np})\right) \tag{3.32}$$

where $h_{\phi_1} : \mathbb{R} \to \mathbb{R}^M$ and $\rho_{\phi_2} : \mathbb{R}^M \to \mathbb{R}$ are DNN mappings with parameters ϕ_1 and ϕ_2, respectively. $\mathbf{s}_{np}$ denotes the couples of observed value y_{np} and corresponding dimension index p. In comparison to the zero imputation method, this framework is theoretically appealing as it makes no assumptions about the values of the missing data - their representations are simply excluded from the summation.

The formulation in (3.32) is identical to the partial VAE framework established in Ma et al. (2019), who, to the best of our knowledge, are the first to consider the use of permutation invariant set functions for handling partially observed data. We begin by considering two specifications of $g_\phi(\mathbf{y}_n^o)$, before considering a closely related partial inference network that factorises across observations.

PointNet

Inspired by the PointNet approach of Qi et al. (2017) and later developed by Ma et al. (2019) for use in partial VAEs, the PointNet specification of partial inference network uses the concatenation of dimension index with the observed value:

$$\mathbf{s}_{np} = (p,\ y_{np})\,. \tag{3.33}$$

Because the dimension indices are input into a continuous function, a central feature of the PointNet is the assumption of smoothness between values of neighbouring dimensions. Although this is often valid in computer vision tasks, it is ill-suited for tasks in which the indexing of dimensions is arbitrary. For example, Ma et al. (2019) consider a task in which the observed values correspond to answers in a questionnaire. Since, in general, questions can be re-ordered whilst representing the same information, the assumption of smoothness across the answers is inappropriate.

IndexNet

An alternative approach to the PointNet specification that places no assumptions on the ordering of observations uses the dimension index to select the encoding function:

$$h_{\phi_1}(\mathbf{s}_{np}) = h_{\phi_{1,p}}(y_{np}). \tag{3.34}$$

Whereas PointNet treats dimension indices as points in space, this specification retains their role as indices. Accordingly, we refer to it as the IndexNet specification of partial inference networks.

Unlike the PointNet specification or zero imputation, not all the variational parameters are amortised across observation dimensions as well as data points. This generally necessitates the use of a greater number of variational parameters which reduces the computational efficiency of the optimisation procedure. For very high-dimensional data the reduction in computational efficiency may become problematic. Further, if few values are observed for any particular dimension then $\phi_{1,p}$ will have little data to be trained on and $h_{\phi_{1,p}}$ may poorly generalise to unseen data[3].

FactorNet

A similar approach to that of IndexNet, first proposed by Vedantam et al. (2017), uses a separate inference network for each observation dimension. Specifically, we factorise the approximate likelihood into a product of Gaussians, one for each observed dimension:

$$l_\phi(\mathbf{f}_n|\mathbf{y}_n^o) = \prod_{p\in\mathcal{O}_n} \underbrace{\mathcal{N}\left(\mathbf{f}_n; \boldsymbol{\mu}_{\phi_p}(y_{np}), \operatorname{diag} \boldsymbol{\sigma}^2_{\phi_p}(y_{np})\right)}_{l_{\phi_p}(\mathbf{f}_n|y_{np})}. \tag{3.35}$$

Each sub-factor $l_{\phi_p}(\mathbf{f}_n|y_{np})$ is parameterised by an inference network specific to dimension p with parameters ϕ_p. Exploiting the properties of exponential distributions, the natural parameters of the resultant Gaussian $l_\phi(\mathbf{f}_n|\mathbf{y}_n^o)$ are found by summing together the natural parameters of each individual Gaussian:

$$\boldsymbol{\eta}_{1,\phi}(\mathbf{y}_n^o) = \sum_{p\in\mathcal{O}_n} \boldsymbol{\eta}_{1,\phi_p}(y_{np}) \quad \text{and} \quad \boldsymbol{\eta}_{2,\phi}(\mathbf{y}_n^o) = \sum_{p\in\mathcal{O}_n} \boldsymbol{\eta}_{2,\phi_p}(y_{np}). \tag{3.36}$$

The transformation from means and variances to natural parameters, and vice-versa, can be achieved using the identities

$$\boldsymbol{\eta}_1 = \frac{\boldsymbol{\mu}}{\boldsymbol{\sigma}^2} \quad \text{and} \quad \boldsymbol{\eta}_2 = -\frac{\mathbf{1}}{2\boldsymbol{\sigma}^2}. \tag{3.37}$$

In keeping with PointNet and IndexNet, we refer to this approach as FactorNet. Observe that FactorNet is equivalent to IndexNet with ρ_{ϕ_2} defined by the deterministic transformations in (3.37). Since IndexNet allows this transformation to be learnt, we can anticipate that it produces a better partial inference network for the task at hand.

The FactorNet approach has the theoretically appealing property that the variance of the approximate likelihood, $l_\phi(\mathbf{f}_n|\mathbf{y}_n^o)$, is guaranteed to decrease as more values are observed. This guarantee is not shared with any of the other three approaches. Furthermore, similar to IndexNet, the method makes no assumptions on the dependencies between observed dimensions. A drawback of FactorNet is that no amortisation takes place across observation dimensions, as

[3] It is worth emphasising that the choice of inference network has no effect on the true posterior. The quality of the optimal approximate posterior for the observed data can only increase with inference network capacity.

each variational parameter belongs to a dimension specific inference network. For relatively high-dimensional datasets, we can expect the FactorNet approach to be the least computationally efficient.

3.4 Sparse Approximations

The computational complexity associated with learning in GP-VAEs is $\mathcal{O}\left(KN^3\right)$. In Section 3.2.3 we saw that a Monte Carlo estimate of the ELBO could be used to reduce this to $\mathcal{O}\left(K\tilde{N}^3\right)$. However, this estimator is biased due to strong correlations between latent variables. Further, it does not reduce the $\mathcal{O}\left(KN^3\right)$ cost of performing inference at test time. Fortunately, there exists a wealth of established sparse GP frameworks that can be leveraged to overcome these limitations. Most notably, the VFE approach of Titsias (2009), introduced in Chapter 2, fits elegantly within the framework of VI already laid out, making it a natural choice for introducing sparse approximations into the GP-VAE.

3.4.1 Sparse Gaussian Processes

Following Matthews et al. (2016), we explicitly parameterise the approximate distribution over inducing points $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_1, \ldots, \mathbf{u}_K)$ whilst retaining the GP conditional prior:

$$q(f) = p_{\theta_1}(f_{\setminus \mathbf{u}}|\mathbf{u})q(\mathbf{u}) = \prod_{k=1}^{K} p_{\theta_1}(f_{k\setminus \mathbf{u}_k}|\mathbf{u}_k)q(\mathbf{u}_k). \tag{3.38}$$

$q(\mathbf{u}_k)$ is restricted to be Gaussian with mean $\mathbf{m}_k$ and covariance $\mathbf{S}_k$, such that each $q(f_k)$ is a GP with mean and covariance function defined by

$$\begin{aligned} m_{q_k}(\mathbf{x}) &= k_{f_k\mathbf{u}_k}\mathbf{K}_{\mathbf{u}_k\mathbf{u}_k}^{-1}\mathbf{m}_k \\ k_{q_k}(\mathbf{x}, \mathbf{x}') &= k_{f_k f_k'} - k_{f_k\mathbf{u}_k}\mathbf{K}_{\mathbf{u}_k\mathbf{u}_k}^{-1}\left(\mathbf{K}_{\mathbf{u}_k\mathbf{u}_k} - \mathbf{S}_k\right)\mathbf{K}_{\mathbf{u}_k\mathbf{u}_k}^{-1}k_{\mathbf{u}_k f_k}. \end{aligned} \tag{3.39}$$

The variational parameters $\phi = \{\mathbf{Z}_k, \mathbf{m}_k, \mathbf{S}_k\}_{k=1}^K$ and model parameters θ are found by maximising the ELBO, given by

$$\begin{aligned} \mathcal{L}_{\text{ELBO}} &= \mathbb{E}_{q(f)}\left[\log \frac{\cancel{p_{\theta_1}(f_{\setminus \mathbf{u}}|\mathbf{u})}p_{\theta_1}(\mathbf{u})p_{\theta_2}(\mathbf{y}|\mathbf{f})}{\cancel{p_{\theta_1}(f_{\setminus \mathbf{u}}|\mathbf{u})}q(\mathbf{u})}\right] = \mathbb{E}_{q(\mathbf{f},\mathbf{u})}\left[\log \frac{p_{\theta_1}(\mathbf{u})p_{\theta_2}(\mathbf{y}|\mathbf{f})}{q(\mathbf{u})}\right] \\ &= \sum_{n=1}^{N}\mathbb{E}_{q(\mathbf{f}_n)}\left[\log p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n)\right] - \sum_{k=1}^{K}\mathrm{KL}\left(q(\mathbf{u}_k) \parallel p_{\theta_{1,k}}(\mathbf{u}_k)\right). \end{aligned} \tag{3.40}$$

Unlike in linear GP models, maximising (3.40) with respect to the set $\{\mathbf{m}_k, \mathbf{S}_k\}_{k=1}^K$ has no analytical solution due to the intractability of the expected non-linear likelihood. Fortunately, the uncollapsed bound lends itself to stochastic optimisation as it decomposes to a sum over individual data points (Hensman et al., 2013).

This approach reduces the computational complexity associated with learning and inference to $\mathcal{O}\left(KNM^2\right)$, where M, the number of inducing points per latent dimension, can be chosen to achieve a desired degree of accuracy. For $M \ll N$, this represents a significant computational saving relative to the GP-VAE. Yet, the model ceases to take the form of a VAE as the approximate posterior is no longer parameterised by an inference network. Indeed, the use of free form variational parameters reintroduces the limitations of mean-field VI: inference cannot be performed on previously unseen data without re-optimising the variational lower bound. Furthermore, in settings in which the size and complexity of the dataset grows sequentially it may be necessary to increase the expressiveness of the approximate posterior through the inclusion of additional inducing points. Doing so does not fit within the existing framework without restarting the optimisation procedure. This can become an insurmountable computational burden for large datasets.

A natural solution is to reintroduce the inference network, using it to parameterise the approximate distribution $q(\mathbf{u})$. It is not immediately obvious how this can be achieved as there is no one-to-one correspondence between inducing points and observations. Fortunately, we can use the ideas introduced in the development of PointNet (Qi et al., 2017) to find a pragmatic workaround.

3.4.2 Sparse GP-VAEs

Recall the structured approximate posterior used in GP-VAEs, repeated here for convenience:

$$q(f) = p_{\theta_1}(f_{\setminus \mathbf{f}}|\mathbf{f}) \frac{1}{\mathcal{Z}_q(\theta, \phi)} p_{\theta_1}(\mathbf{f}) l_\phi(\mathbf{f}|\mathbf{y}).$$

Sparseness can be introduced into the approximate posterior by replacing the approximate likelihood, $l_\phi(\mathbf{f}|\mathbf{y})$, with an approximate likelihood over inducing points, $l_\phi(\mathbf{u}|\mathbf{y})$. The corresponding approximate posterior takes the form

$$q(f) = p_{\theta_1}(f_{\setminus \mathbf{u}}|\mathbf{u}) \frac{1}{\mathcal{Z}_q(\theta, \phi)} \underbrace{p_{\theta_1}(\mathbf{u}) l_\phi(\mathbf{u}|\mathbf{y})}_{q(\mathbf{u})} \tag{3.41}$$

where the approximate likelihood factorises across data points, latent dimensions and inducing points:

$$l_\phi(\mathbf{u}|\mathbf{y}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \prod_{m=1}^{M} l_\phi(u_{mk}|\mathbf{y}_n). \tag{3.42}$$

A distinguishing feature of the approximate likelihood over inducing points, $l_\phi(u_{mk}|\mathbf{y}_n)$, is that it conditions on data at locations different to those of the inducing points. For stationary kernels, the strength of the dependence exhibited by latent function values is determined by the difference between the inputs. Since there is a one-to-one correspondence between

latent function values and observations, this dependency extends to the relationship between observations and latent function values. Employing the standard inference network, which conditions solely on the observed value $\mathbf{y}_n$, does not take this into account, and is therefore ill-suited for parameterising $l_\phi(u_{mk}|\mathbf{y}_n)$. Rather, the output of the inference network must depend on the difference between the location of each observation and the location of the inducing point whose approximate likelihood is being parameterised. In the spirit of machine learning, this can be achieved by treating the difference between the input locations as inputs to the inference network and letting the network learn the conditioning that best approximates the true posterior.

Formally, let $\mathbf{z}_{mk}$ denote the location of inducing point u_{mk} and $\mathbf{x}_n$ denote the location of observation $\mathbf{y}_n$. For each observation/inducing point pair $(u_{mk}, \mathbf{y}_n)$, the modified inference network maps from $(\mathbf{z}_{mk} - \mathbf{x}_n, \mathbf{y}_n)$ to parameters of an approximate likelihood factor $l_\phi(u_{mk}|\mathbf{y}_n, \mathbf{z}_{mk}, \mathbf{x}_n)$:

$$l_\phi(\mathbf{u}|\mathbf{y}, \mathbf{Z}, \mathbf{X}) = \prod_{n=1}^{N}\prod_{k=1}^{K}\prod_{m=1}^{M} \underbrace{\mathcal{N}\left(u_{mk}; \mu_\phi(\mathbf{z}_{mk} - \mathbf{x}_n, \mathbf{y}_n), \sigma^2_\phi(\mathbf{z}_{mk} - \mathbf{x}_n, \mathbf{y}_n)\right)}_{l_\phi(u_{mk}|\mathbf{y}_n, \mathbf{z}_{mk}, \mathbf{x}_n)}. \tag{3.43}$$

Note the similarity between this approach and that of PointNet (Qi et al., 2017). Indeed, its application here is more suitable than its use in handling partially observed data, as the assumption of continuity in the dependence of the output on $\mathbf{z}_{mk} - \mathbf{x}_n$ is appropriate. For each inducing point there will be N approximate likelihood factors, one for each observation, which can be combined into a single approximate likelihood through the addition of natural parameters. Observing that $\mathbf{u}$ replaces the role of $\mathbf{f}$ in the structured approximate posterior for the GP-VAE, the ELBO is simply

$$\mathcal{L}_{\text{ELBO}} = \sum_{n=1}^{N} \mathbb{E}_{q(\mathbf{f}_n)}\left[\log p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n)\right] - \mathbb{E}_{q(\mathbf{u})}\left[\log l_\phi(\mathbf{u}|\mathbf{y})\right] + \log \mathcal{Z}_q(\theta, \phi) \tag{3.44}$$

where

$$\mathcal{Z}_q(\theta, \phi) = \prod_{k=1}^{K} \int p_{\theta_{1,k}}(\mathbf{u}_k) l_\phi(\mathbf{u}_k|\mathbf{y}) d\mathbf{u}_k. \tag{3.45}$$

Sampling from $q(\mathbf{f}_n)$ requires the mean and covariance functions of the approximate GP, given by

$$\hat{m}_k(\mathbf{x}) = k_{f_k \mathbf{u}_k}\left(\mathbf{K}_{\mathbf{u}_k\mathbf{u}_k} + \mathbf{\Sigma}_{\phi,k}\right)^{-1} \boldsymbol{\mu}_{\phi,k} \tag{3.46}$$

$$\hat{k}_k(\mathbf{x}) = k_{f_k f_k'} - k_{f_k \mathbf{u}_k}\left(\mathbf{K}_{\mathbf{u}_k\mathbf{u}_k} + \mathbf{\Sigma}_{\phi,k}\right)^{-1} k_{\mathbf{u}_k f_k} \tag{3.47}$$

where $\boldsymbol{\mu}_{\phi,k}$ and $\mathbf{\Sigma}_{\phi,k}$ denote the mean and diagonal covariance of $l_\phi(\mathbf{u}_k|\mathbf{y})$. We refer to the use of this approximate posterior as the sparse GP-VAE.

For the same number of inducing points, the optimal approximate posterior of the sparse GP-VAE is strictly worse than the optimal free-form sparse approximate posterior. However, the inference network can be used to condition on previously unobserved data without needing to learn new variational parameters. Moreover, the inference network can be shared across datasets of similar characteristics, which can significantly reduce the total number of variational parameters and thus the computational complexity of the optimisation procedure. A potential limitation of this approach is that, when a large number of inducing points are used, having to make forward and backward passes through the inference network NM times can become costly. This can be circumvented by requiring the approximate distribution for only the T most correlated[4] inducing points to be specified by each observation, requiring a total of NT passes through the inference network.

It should be emphasised that the sparse GP-VAE places no restrictions on the inducing point locations - these are also variational parameters of the approximate posterior. The practitioner is given complete autonomy over the initialisation of locations: in the temporal setting, it is common to initialise inducing points at fixed intervals throughout the input domain; in the more general spatio-temporal setting, inducing points are often initialised using random selection, or k-means clustering, on the input locations included in the observed data.

Not only does the sparse GP-VAE address the issue of performing inference on previously unseen data, it also addresses the second major shortcoming of existing sparse GP frameworks, namely that the complexity of the approximate posterior cannot be increased without restarting the optimisation procedure. For the sparse GP-VAE, inducing points can be added, moved around or removed as desired without needing to perform any additional training, let alone restart the optimisation procedure. The inference network places no restrictions on the morphology of the inducing points; it simply learns to map from observations to approximate likelihoods that produce good posterior approximations. For example, in settings in which data is being streamed through time, inducing points can be introduced sequentially to model the regions of newly observed data without any additional training. We anticipate the implications of the sparse GP-VAE to be widespread.

[4]For SE kernels, this is the nearest T inducing points. For periodic kernels, this is the nearest T inducing points after a sinusoidal transformation.

4 | Related Work

In this chapter, we provide an overview of, and make connections to, the existing literature that shares common themes with the GP-VAE. Section 4.1 begins with a review of existing approaches to the use of structured latent priors in VAEs, notably demonstrating that the GP-VAE is a special case of Johnson et al.'s (2016) structured VAE. A comparison between our approach and previous attempts at integrating GPs into the VAE framework is also made. Section 4.2 establishes a unifying connection between the probabilistic model employed by the GP-VAE and other multi-output GPs, including the family of linear multi-output GPs and deep GPs. Finally, Section 4.3 compares the approximate inference framework employed by the GP-VAE with expectation propagation.

4.1 Structured Priors in Variational Autoencoders

Structured Variational Autoencoders

Only recently has the use of structured latent variable priors in VAEs been considered. In their seminal work, Johnson et al. (2016) investigate the combination of probabilistic graphical models with neural networks to learn structured latent variable representations with flexible likelihoods. The authors consider the general case of a prior composed of a conjugate pair of exponential family distributions over global latent variables, θ, and local latent variables $\mathbf{z} = \{\mathbf{z}_n\}_{n=1}^N$:

$$\begin{aligned} p(\theta) &= \exp\left\{\eta_\theta^{0T} t_\theta(\theta) - \log \mathcal{Z}_\theta(\eta_\theta^0)\right\} \\ p(\mathbf{z}|\theta) &= \exp\left\{\eta_\mathbf{z}^0(\theta)^T t_\mathbf{z}(\mathbf{z}) - \log \mathcal{Z}_\mathbf{z}(\eta_\mathbf{z}^0)\right\} \end{aligned} \tag{4.1}$$

with a likelihood function $p(\mathbf{y}|\mathbf{z})$. The conjugate exponential family latent variable model has greater flexibility than the standard normal prior, encompassing priors such as linear dynamical systems (LDS) and Gaussian mixture models (GMM).

To circumvent the issue of an intractable posterior, Johnson et al. introduce a factorised approximate posterior, $q(\theta, \mathbf{z}) = q(\theta)q(\mathbf{z})$. The ELBO is given by

$$\mathcal{L}_{\text{ELBO}}(\eta_\theta, \eta_\mathbf{z}) = \mathbb{E}_{q(\theta)q(\mathbf{z})}\left[\log \frac{p(\theta)p(\mathbf{z}|\theta)p(y|\mathbf{z},\theta)}{q(\theta)q(\mathbf{z})}\right], \tag{4.2}$$

where η_θ and $\eta_\mathbf{z}$ denote the natural parameters of $q(\theta)$ and $q(\mathbf{z})$, respectively. In the general case of a non-conjugate likelihood, existing frameworks for performing efficient approximate inference in graphical models[1] cannot be used. Instead, Johnson et al. introduce a surrogate objective $\hat{\mathcal{L}}$

$$\hat{\mathcal{L}}(\eta_\theta, \eta_\mathbf{z}, \phi) \coloneqq \mathbb{E}_{q(\theta)q(\mathbf{z})}\left[\log \frac{p(\theta)p(\mathbf{z}|\theta)\exp\{\psi(\mathbf{z};\mathbf{y},\phi)\}}{q(\theta)q(\mathbf{z})}\right], \quad \psi(\mathbf{z};\mathbf{y},\phi) \coloneqq r(\mathbf{y};\phi)^T t_\mathbf{z}(\mathbf{z}) \tag{4.3}$$

where $r(\mathbf{y};\phi)$ denotes the output of the inference network with parameters ϕ. The inference network maps from the observations to the natural parameters of a 'pseudo likelihood' in a 'pseudo graphical model'. Crucially, unlike the true likelihood, the pseudo likelihood is conjugate to the latent prior. The previously unusable approximate inference frameworks can now be applied to efficiently obtain the local optimiser $q^*(\mathbf{z})$ with natural parameters

$$\eta^*_\mathbf{z}(\eta_\theta, \phi) \coloneqq \arg\max_{\eta_\mathbf{z}} \hat{\mathcal{L}}. \tag{4.4}$$

Substituting $q^*(\mathbf{z})$ into the original ELBO defines the structured VAE (SVAE) objective:

$$\mathcal{L}_{\text{SVAE}}(\eta_\theta, \phi) \coloneqq \mathcal{L}_{\text{ELBO}}(\eta_\theta, \eta^*_\mathbf{z}(\eta_\theta, \phi)), \tag{4.5}$$

which can be optimised with respect to η_θ and ϕ using gradient based methods. This process is iterated until convergence.

In the case of fixed global latent variables θ, the surrogate objective becomes

$$\hat{\mathcal{L}} = \mathbb{E}_{q(\mathbf{z})}\left[\log \frac{p_{\theta_1}(\mathbf{z})\exp\{\psi(\mathbf{z};\mathbf{y},\phi)\}}{q(\mathbf{z})}\right] \tag{4.6}$$

which is maximised by the local optimiser

$$q^*(\mathbf{z}) = \frac{1}{\mathcal{Z}_q(\phi,\theta)} p_{\theta_1}(\mathbf{z})\exp\{\psi(\mathbf{z};\mathbf{y},\phi)\}. \tag{4.7}$$

[1] Including expectation propagation (Minka, 2001), variational message passing (Winn and Bishop, 2005) and stochastic VI (Hoffman et al., 2013).

The corresponding SVAE objective is given by

$$\mathcal{L}_{\text{SVAE}}(\phi, \theta) = \mathbb{E}_{q^*(\mathbf{z};\phi,\theta)}\left[\log \frac{p_{\theta_1}(\mathbf{z})p_{\theta_2}(\mathbf{y}|\mathbf{z})}{\frac{1}{\mathcal{Z}_q(\phi,\theta)}p_{\theta_1}(\mathbf{z})\exp\{\psi(\mathbf{z};\mathbf{y},\phi)\}}\right]. \tag{4.8}$$

This is equivalent to optimising the ELBO using the structured approximate posterior

$$q(\mathbf{z}) = \frac{1}{\mathcal{Z}_q(\phi,\theta)}p_{\theta_1}(\mathbf{z})l_\phi(\mathbf{z}|\mathbf{y}), \tag{4.9}$$

where $l_\phi(\mathbf{y}|\mathbf{z}) = \exp\{\psi(\mathbf{z};\mathbf{y},\phi)\}$. Choosing $p_{\theta_1}(\mathbf{z})$ to be defined by a GP recovers the GP-VAE.

Structured Inference Networks

Lin et al. (2018) build upon the SVAE, proposing a structured approximate posterior of the form

$$q(\mathbf{z}) = \frac{1}{\mathcal{Z}_q(\phi)}q_{\phi_{\text{PGM}}}(\mathbf{z})l_{\phi_{\text{NN}}}(\mathbf{z}|\mathbf{y}). \tag{4.10}$$

The authors refer to the approximate posterior as the structured inference network (SIN). Rather than using the latent prior $p_{\theta_1}(\mathbf{z})$, SIN incorporates the model's latent structure through $q_{\phi_{\text{PGM}}}(\mathbf{z})$. The core advantage of SIN is its extension to more complex latent priors containing non-conjugate factors - $q_{\phi_{\text{PGM}}}(\mathbf{z})$ can replace them with their nearest conjugate approximations whilst retaining a similar latent structure.

Whilst the frameworks proposed by Johnson et al. and Lin et al. are more general than ours, in both cases the authors only consider GMM and LDS latent priors. Priors with stronger dependencies, such as GPs, are neglected for the sake of computational efficiency.

Gaussian Process Priors

We are not the first to develop inference techniques in VAEs with a GP prior. To the best of our knowledge, the earliest example is the GP prior VAE (Casale et al., 2018). There are significant differences between our work and Casale et al.'s, most notably in their use of a fully-factorised approximate posterior and low-rank approximation of the GP prior. Further, the authors only consider the case of fully observed data, neglecting to account for the presence of partial observations. Fortuin et al. (2020) consider a generative model identical to ours; however, they employ a Gaussian approximate posterior with a tridiagonal precision matrix $\mathbf{\Lambda}$:

$$q(\mathbf{z}) = \mathcal{N}\left(\mathbf{z}; \mathbf{m}, \mathbf{\Lambda}^{-1}\right) \tag{4.11}$$

where

$$\mathbf{\Lambda} \coloneqq \mathbf{B}^T\mathbf{B}, \quad [\mathbf{B}]_{ij} = \begin{cases} b_{ij} & \text{if } j \in \{i, i+1\} \\ 0 & \text{otherwise} \end{cases}. \tag{4.12}$$

Inference is amortised using an inference network, g_ϕ, that maps from observations $\mathbf{y}$ to the parameters of the approximate posterior, $(\mathbf{m}, \mathbf{B}) = g_\phi(\mathbf{y})$. The convenience of this parameterisation is that $q(\mathbf{z})$ can be sampled from with linear computational complexity rather than the cubic computational complexity associated with a full-rank covariance matrix. This efficiency comes at the sacrifice of versatility and modelling rigour. The parameterisation is only appropriate for regularly spaced temporal data and neglects rigorous treatment of long term dependencies. Campbell and Liò (2020) employ an equivalent sparsely structured variational posterior as Fortuin et al.'s (2020), extending the framework to handle more general spatio-temporal data. Their method is similarly restricted to regularly spaced spatio-temporal data. A fundamental difference between our framework and that of Fortuin et al. and Campbell and Liò is the inclusion of the GP prior in the approximate posterior. Neglecting the prior neglects the very dependencies that characterise spatio-temporal datasets.

Most similar to ours is the approach of Pearce (2020), which also employs a similarly structured approximate posterior. However, the author only considers the application to modelling pixel dynamics. This work is the first to develop the use of partial inference networks and sparse approximations in the GP-VAE.

4.2 Multi-Output Gaussian Processes

Through consideration of the interchange of input dependencies and likelihood functions, we can shed light on the relationship between the probabilistic model employed by the GP-VAE and other multi-output GP models. These relationships are summarised in Figure 4.1.

Linear Multi-Output Gaussian Processes

Replacing the likelihood with a linear likelihood function characterises a family of linear multi-output GPs, defined by a linear transformation of K independent latent GPs:

$$\begin{aligned} f &\sim \prod_{k=1}^{K} \mathcal{GP}\left(0, k_{\theta_{1,k}}(\mathbf{x}, \mathbf{x}')\right) \\ \mathbf{y}|f &\sim \prod_{n=1}^{N} \mathcal{N}\left(\mathbf{y}_n; \mathbf{W}\mathbf{f}_n, \mathbf{\Sigma}\right). \end{aligned} \tag{4.13}$$

The family includes Teh et al.'s (2005) semiparametric latent factor model (SLFM)[2], Byron et al.'s (2009) GP factor analysis (GP-FA) and Bonilla et al.'s (2008) class of multi-task GPs. Notably, removing input dependencies by choosing $k_{\theta_{1,k}}(\mathbf{x}, \mathbf{x}') = \delta(\mathbf{x}, \mathbf{x}')$ recovers factor

[2]The name 'semiparametric latent factor model' arises due to the combination of a nonparametric latent prior (the GP) and parametric likelihood function. Whilst Teh et al. only considered the case of a linear likelihood, one could view the probabilistic model employed by the GP-VAE as a direct extension to their work in which the parametric likelihood function is a DNN.

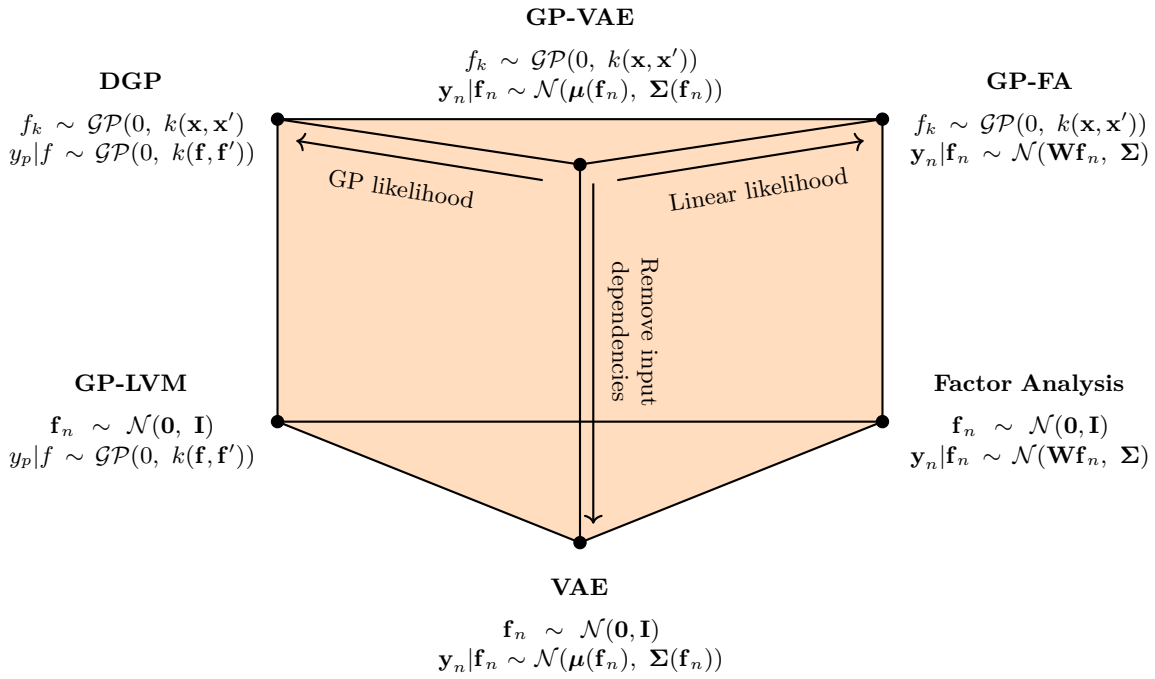


Fig. 4.1 A unifying perspective on multi-output GPs.

analysis, or equivalently, probabilistic principal component analysis (Tipping and Bishop, 1999) when $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$. Akin to the relationship between factor analysis and linear multi-output GPs, standard VAEs can be viewed as a special, instantaneous case of GP-VAEs.

Deep Gaussian Processes

Single layered deep GPs (DGPs) (Damianou and Lawrence, 2013) are characterised by the use of a GP likelihood function[3], giving rise to the generative model

$$\begin{aligned} f &\sim \prod_{k=1}^{K} \mathcal{GP}\left(0, k_{\theta_{1,k}}(\mathbf{x},\mathbf{x}')\right) \\ y|f &\sim \prod_{p=1}^{P} \mathcal{GP}\left(0, k_{\theta_{2,p}}(f(\mathbf{x}), f(\mathbf{x}'))\right) \end{aligned} \tag{4.14}$$

where $\mathbf{y}_n = y(\mathbf{x}_n)$. The GP latent variable model (GP-LVM) (Lawrence and Moore, 2007) is the special, instantaneous case of single layered DGPs. Multi-layered DGPs are recovered using a hierarchical latent space with conditional GP priors between each layer.

[3] Adopting the naming convention of Teh et al. (2005), an appropriate name for this general class of models is 'nonparametric latent factor models'.

4.3 Expectation Propagation

The GP-VAE employs the structured approximate posterior

$$q(\mathbf{f}) = \frac{1}{\mathcal{Z}_q(\theta, \phi)} p_{\theta_1}(\mathbf{f}) \prod_{n=1}^{N} l_\phi(\mathbf{f}_n|\mathbf{y}_n) \tag{4.15}$$

where the approximate likelihoods $l_\phi(\mathbf{f}_n|\mathbf{y}_n)$ are parameterised by an inference network. For the case in which the parameters of $l_\phi(\mathbf{f}_n|\mathbf{y}_n)$ are freely optimisable, the approximate posterior used in expectation propagation (EP) is recovered (Minka, 2001). Recently, considerable success has been found using EP to unify sparse approximation schemes in GP and DGP models (Bui, 2018). EP approximates the GP posterior using the unnormalised structured approximate posterior

$$q(f) \propto p_\theta(f) \prod_{n=1}^{N} l_{\phi_n}(\mathbf{u}). \tag{4.16}$$

Whilst VI uses gradient based methods to minimise the global exclusive KL divergence $\mathrm{KL}\left(q(f) \parallel p_\theta(f|\mathbf{y})\right)$, EP employs an iterative fixed point procedure that considers the minimisation of the local inclusive KL divergence

$$\phi_n \leftarrow \arg\min_{\phi_n} \mathrm{KL}\left(q_{\backslash n}(f) p(y_n|f_n) \parallel q_{\backslash n}(f) l_{\phi_n}(\mathbf{u})\right) \tag{4.17}$$

where $q_{\backslash n} \propto \frac{q(f)}{l_{\phi_n}(\mathbf{u})}$ is the 'cavity distribution'. The procedure refines each $l_{\phi_n}(\mathbf{u})$ to approximate the contribution of each local likelihood $p_{\theta_2}(\mathbf{y}_n|\mathbf{f}_n)$ to the true posterior. The algorithmic differences between VI and EP have several important consequences:

Approximate posterior: EP minimises the inclusive KL divergence, whereas VI minimises the exclusive KL divergence. Thus, we can expect the approximate posterior found using EP to avoid the mode matching behaviour of VI and, in turn, overconfident approximations. This is not always a positive characteristic: in the case of bi-modal likelihood factors, minimising the inclusive KL divergence with a uni-modal $l_{\phi_n}(\mathbf{u})$ will incorrectly assign high probability to regions of low probability.

Computational efficiency: EP is often considerably slower than VI. This owes to the difference in optimisation procedures. Specifically, VI can employ the wealth of stochastic optimisation procedures to significantly improve efficiency. Although stochastic methods for EP exist (Li et al., 2015; Vehtari et al., 2014), these place stronger restrictions on the approximate posterior. Furthermore, simultaneously optimising model parameters and variational parameters does not fit naturally within the EP framework. It is common practice to tune model parameters after the procedure has converged, requiring it to be run again.

Amortisation: EP requires the parameters of the approximate likelihoods to be tuned individually. When the approximate likelihoods are parameterised by an inference network, this is impossible. Thus, EP shares the same limitations as mean-field VI.

5 | Experiments

In this chapter we detail an experimental investigation into the modelling capability of the GP-VAE. Section 5.1 begins with a discussion of the key implementation details that were found to improve the efficiency and robustness of the GP-VAE. Section 5.2 evaluates the relative performance of the ELBO estimators derived in Chapter 3. In Sections 5.3 to 5.5, we demonstrate the capability of the GP-VAE on a wide range of experiments involving spatio-temporal datasets with distinguishable characteristics, comparing its performance to other multi-output GP models and structured VAEs. Finally, Section 5.6 probes the performance of the GP-VAE and sparse GP-VAE on a large spatio-temporal weather dataset.

The Python implementation of the GP-VAE as well as all the experiments conducted in this thesis can be found at https://github.com/MattAshman/SpatioTemporalVAE.

5.1 Implementation Details

Whilst the techniques outlined in this section are not strictly necessary, they often help to avoid numerical instabilities, improve computational efficiency and improve the performance of the GP-VAE.

5.1.1 Avoiding Numerical Instabilities

A regularly encountered problem when implementing GP based models are numerical instabilities arising when trying to invert poorly conditioned matrices. In particular, training and testing the GP-VAE requires sampling each of the K approximate posterior GPs with mean and covariance functions taking the form

$$\begin{aligned} \hat{m}(\mathbf{x}) &= k_{f\mathbf{f}} \left(\mathbf{K}_{\mathbf{ff}} + \mathbf{\Sigma}_\phi\right)^{-1} \boldsymbol{\mu}_\phi \\ \hat{k}(\mathbf{x}, \mathbf{x}') &= k_{ff'} - k_{f\mathbf{f}} \left(\mathbf{K}_{\mathbf{ff}} + \mathbf{\Sigma}_\phi\right)^{-1} k_{\mathbf{f}f}. \end{aligned} \tag{5.1}$$

Note that we have dropped the subscript k for notational convenience. $\left(\mathbf{K}_{\mathbf{ff}} + \mathbf{\Sigma}_\phi\right)^{-1}$ may be extremely poorly conditioned for large N, especially when the variances of the approximate

posteriors (the elements of $\mathbf{\Sigma}_\phi$) are small or the lengthscales of the GP priors are large. The inversion $(\mathbf{K_{ff}} + \mathbf{\Sigma}_\phi)^{-1}$ can be avoided by drawing out the Cholesky factor $\mathbf{\Sigma}_\phi^{-\frac{1}{2}}$:

$$(\mathbf{K_{ff}} + \mathbf{\Sigma}_\phi)^{-1} = \mathbf{\Sigma}_\phi^{-\frac{1}{2}} \left(\mathbf{\Sigma}_\phi^{-\frac{1}{2}} \mathbf{K_{ff}} \mathbf{\Sigma}_\phi^{-\frac{1}{2}} + \mathbf{I} \right)^{-1} \mathbf{\Sigma}_\phi^{-\frac{1}{2}}. \tag{5.2}$$

The eigenvalues of $\mathbf{\Sigma}_\phi^{-\frac{1}{2}} \mathbf{K_{ff}} \mathbf{\Sigma}_\phi^{-\frac{1}{2}} + \mathbf{I}$ are bounded below by 1 and above by $1 + \frac{N}{4} \max_{ij} \left([\mathbf{K_{ff}}]_{ij} \right)$ (Rasmussen and Williams, 2005). To improve numerical stability and computational efficiency further, matrix inversions are performed using the Cholesky decomposition.

We found that the covariance $\mathbf{K_{ff}}$ often became numerically singular for large N. In such cases, adding a small amount of jitter to the diagonal can restore positive definiteness - we used 10^{-5} to ensure numerical stability in all our experimentation. Compared to the overall stochasticity of the Monte Carlo estimators, the effect is negligible.

5.1.2 Avoiding Posterior Collapse

During training, we observed that for some latent dimensions the variance of the approximate likelihood occasionally grew very large in concurrence with the scale of the GP prior collapsing to near zero. This removes any dependence on the observed data from these posterior GPs, constraining the effective approximate posterior to fewer latent dimensions than specified. We found this to result in a generally poorer fit to the observed data. To understand the origin of this effect, recall the decomposition of the ELBO

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q(\mathbf{f})} \left[p_{\theta_2}(\mathbf{y}|\mathbf{f}) \right] - \text{KL} \left(q(\mathbf{f}) \parallel p_{\theta_1}(\mathbf{f}) \right). \tag{5.3}$$

When the variance of the prior is much smaller than that of the approximate likelihood, the structured approximate posterior becomes equal to the prior resulting in the KL term of (5.3) collapsing to zero. Put more succinctly, the approximate posterior gives up on conditioning on the observed data in favour of remaining close to the prior.

We are not the first to observe this phenomenon, more commonly known as 'posterior collapse'. A popular workaround is to warm-up the KL term using the modified ELBO

$$\mathcal{L}^{\beta}_{\text{ELBO}} = \mathbb{E}_{q(\mathbf{f})} \left[\log p_{\theta_2}(\mathbf{y}|\mathbf{f}) \right] - \beta \text{KL} \left(q(\mathbf{f})) \parallel p_{\theta_1}(\mathbf{f}) \right) \tag{5.4}$$

where β is gradually increased from 0 to 1 during training (Bowman et al., 2016; Cremer et al., 2018; Sønderby et al., 2016). Whilst working from a pragmatic perspective, the approach requires a seemingly arbitrary modification to the otherwise theoretically principled ELBO. We found a more effective resolution was to initialise the variance of the approximate likelihoods and true likelihoods to be very small (10^{-4}), with no change to the ELBO necessary. This

strongly encourages the model towards conditioning on, and fitting, the observed data, rather than collapsing to the prior.

5.1.3 Experimental Details

Whilst the theory outlined in Chapters 2 and 3 describes a general decoder parameterising both the mean and variance of the likelihood, we experienced difficulty training GP-VAEs using a learnt variance, especially for high-dimensional observations. Thus, for the experiments detailed in this chapter we use a shared variance across all observations.

We use the Adam optimiser (Kingma and Ba, 2014) with a constant learning rate of 0.001. Unless stated otherwise, we estimate the gradients of the ELBO using a single sample and the ELBO itself using 100 samples. For each experiment, we normalise the observations using the means and standard deviations of the data in the training set.

5.2 Comparing Estimators

In Chapter 3, four methods for estimating the gradients of the ELBO for the GP-VAE were presented: the semi-analytic score function estimator (SA-SF), the doubly-stochastic score function estimator (DS-SF), the semi-analytic path derivative estimator (SA-PD) and the doubly-stochastic path derivative estimator (DS-PD). To compare the effectiveness of each, we consider the task of modelling a toy dataset composed of samples from four interdependent functions:

$$
\begin{aligned}
y_1(x) &= 2\sin(0.2x) - 0.5\cos(x) \\
y_2(x) &= 2\cos(0.5x) + 2\sin(0.1x) - 5 \\
y_3(x) &= -y_1(x) + 3y_2(x) \\
y_4(x) &= 0.5y_1(x) - 2y_2(x).
\end{aligned}
$$

In each case, a two-dimensional latent space with SE kernels is used. All lengthscales and output scales are initialised to 1 and all DNNs are composed of two hidden layers of 20 units with rectified linear unit (ReLU) activation functions. A dataset of 100 data points is formed by sampling each of the four functions uniformly in the range $x \in [-50, 150]$.

5.2.1 Comparing Gradient Estimators

Figure 5.1 compares the progression of the ELBO for GP-VAEs optimised using each of the four estimators. After 5000 epochs, the converged ELBO for the DS-PD estimator is -19.17, notably better than the converged ELBO for the other three estimators, all of which are less than -110. The rate of convergence of the score function estimators is significantly slower than that of the

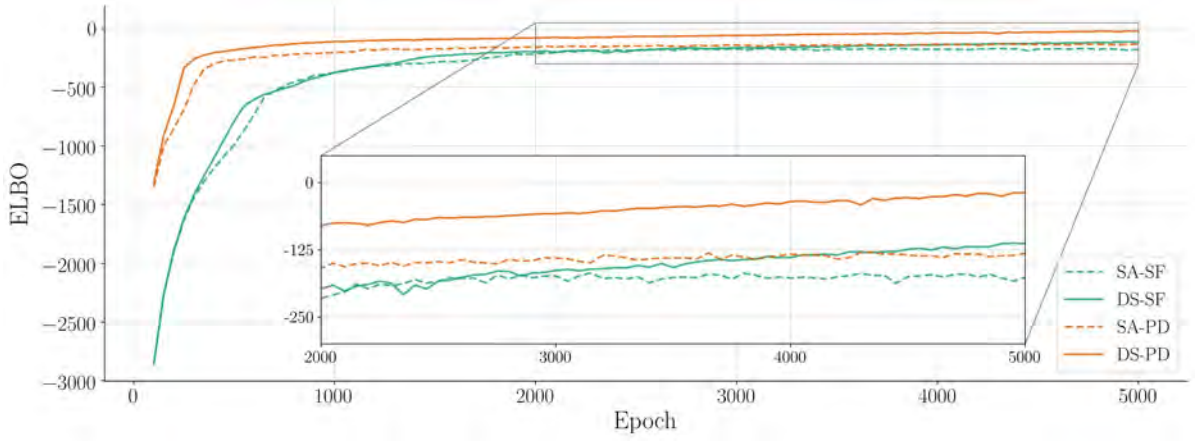


Fig. 5.1 A comparison between the GP-VAE ELBO when trained using each of the four estimators: semi-analytic score function (SA-SF), doubly-stochastic score function (SF), semi-analytic path derivative (SA-PD) and doubly-stochastic path derivative (PD).

path derivative estimators, with the two approaches reaching an ELBO of around -800 and -200 after 50 epochs, respectively. Throughout the training regime, the semi-analytic estimators perform worse than their doubly-stochastic counterparts. As previously discussed in Chapter 3, this suggests that stochasticity of the Monte Carlo estimates cancels, which in turn, suggests that the quality of the approximate posterior is good. Whilst the path derivative estimators perform better than the score function estimators for the majority of training, the DS-SF estimator converges to a larger ELBO than the SA-PD estimator. However, any conclusions made from Figure 5.1 should be treated with caution: the effects of random initialisation can be large, especially when using likelihood functions parameterised by DNNs. Furthermore, because the dataset being modelled is relatively uncomplicated, the learnt likelihood function is likely to be simple and thus easily approximated. It may be that for more complex data, the learnt likelihood is not so easily approximated, in which case, the stochasticity of the doubly-stochastic estimators would not cancel to the same degree. In practice, we found it beneficial to train models using both the DS-PD and SA-PD estimators and evaluate the performance of the model with the highest ELBO.

5.2.2 Comparing ELBO Estimators

Figure 5.2 compares the empirical variance of the semi-analytic ELBO estimator (SA-ELBO) and doubly-stochastic ELBO (DS-ELBO) estimator using a single sample. Similar to 5.1, we see that the doubly-stochastic estimator has a stronger empirical performance than its semi-analytic counterpart, consistent with the claim that the quality of the approximate posterior is good. Observe that the difference between the variance of the estimators sharply increases in synchrony with the increase in ELBO shown in Figure 5.1. The increase in ELBO is due

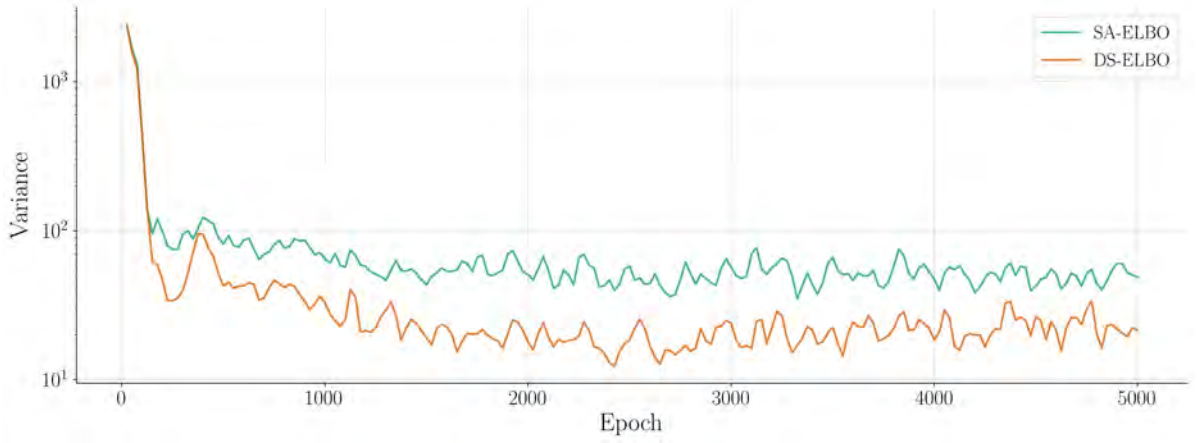


Fig. 5.2 A comparison between the variance of the semi-analytic ELBO estimator (SA-ELBO) and doubly-stochastic ELBO estimator (DS-ELBO) for the GP-VAE trained on the toy dataset using the DS-PD estimator.

to the approximate posterior improving, which results in a reduction in stochasticity of the doubly-stochastic estimators.

5.3 Electroencephalogram Dataset

The electroencephalogram (EEG) dataset was collected for the purpose of examining genetic predisposition to alcohol[1]. The complete dataset consists of 120 recordings for each of the 122 subjects. Each recording contains $N = 256$ measurements recorded over a duration of 1 second, with each measurement being composed of voltage readings taken by 64 electrodes positioned at different locations on a patient's scalp.

Adopting the experimental procedure laid out by Requeima et al. (2019), we consider the smaller dataset of a single, partial recording. Specifically, the dataset is formed using measurements from the seven frontal electrodes FZ and F1-F6 from the first trial on control subject 337. The task is to predict the final 100 samples for electrodes FZ, F1 and F2 having observed the first 156 samples as well as all 256 samples for electrodes F3-F6. Following Requeima et al., we evaluate the model using the standardised mean squared error (SMSE) and negative log-likelihood (NLL). As the final 100 samples are partially observed, the use of a partial inference network is required. In Chapter 3 four candidates were discussed: zero imputation (ZI), PointNet, IndexNet and FactorNet. We evaluate the performance of each.

For all GP-VAE models we use a three-dimensional latent space, each using SE kernels with lengthscales and scales initialised to 0.1 and 1, respectively. Using fewer latent dimensions than observation dimensions forces the model to share information across dimensions, which is

[1]The data can be found at https://archive.ics.uci.edu/ml/datasets/eeg+database.

needed for data imputation tasks such as this. All DNNs, except for those in PointNet and IndexNet, use two hidden layers of 20 units with ReLU activation functions. PointNet and IndexNet employ DNNs with a single hidden layer of 20 units and a 20-dimensional intermediate representation. All GP-VAE models are trained for 3000 epochs using a batch size of 100. The procedure is repeated 15 times, and the mean ± standard deviation of the performance metrics for the 10 iterations with the highest ELBO is reported[2]. Table 5.1 details the results. The performance of the GP-VAE models is compared to that of independent GPs (IGP), the Gaussian process autoregressive regression model (GPAR) (Requeima et al., 2019) and a GP-VAE using a linear likelihood function and IndexNet (GP-VAE-LL).

Table 5.1 A comparison between multi-output GP models on the EEG data imputation task.

				GP-VAE			
Metric	IGP†	GPAR†	GP-VAE-LL	ZI	PointNet	IndexNet	FactorNet
SMSE	1.75	0.26	0.279 ± 0.017	0.272 ± 0.030	0.602 ± 0.088	**0.238 ± 0.019**	0.278 ± 0.043
NLL	2.60	**1.63**	1.898 ± 0.053	2.236 ± 0.367	3.030 ± 1.341	2.012 ± 0.283	2.228 ± 0.210

†Results taken directly from Requeima et al. (2019).

Table 5.1 shows that all multi-output GP methods achieve significantly better predictive performance than independent GPs. This demonstrates the importance of modelling dependencies between voltage readings and not solely temporal dependencies. Amongst multi-output GPs, the GP-VAE using IndexNet achieves a new state-of-the-art average SMSE of 0.238, marginally outperforming both the GP-VAE using zero imputation and FactorNet as well as GPAR, which achieves an average SMSE of 0.26. The average NLLs of the GP-VAEs are noticeably worse than that of GPAR, for which there are two possible explanations:

1. the GP-VAE overfits to the training data, a consequence of the large number of model parameters including in the likelihood function and selection of models with the highest ELBO;
2. the approximate posterior is overconfident, a consequence of employing VI with a Gaussian approximate distribution.

GPAR has few model parameters and performs exact inference, meaning the posterior uncertainty estimates are likely to be better than the GP-VAEs. Nonetheless, the relatively strong performance of the GP-VAE demonstrates its ability to model dependencies between observations and produce accurate posterior predictions in the presence of partially observed

[2]We found that the GP-VAE occasionally got stuck in very poor local optima. Since the ELBO is calculated on the training set alone, the experimental procedure is still valid.

data. It is illuminating to compare the performance of the GP-VAE using linear likelihood[3] to the GP-VAE using a non-linear likelihood. The use of a non-linear likelihood leads to a substantial improvement in average SMSE, yet a worsening in average NLL. The former highlights the benefits of using a more flexible likelihood function on even a relatively simple dataset; however, the latter provides further evidence that the GP-VAE overfits to the training data.

Amongst the GP-VAE models, the use of PointNet results in the worst average SMSE and NLL of 0.602 and 2.519, respectively. Figure 5.3 compares the posterior predictive distributions for two GP-VAE models using IndexNet and PointNet, from which the inferiority when using PointNet is immediately clear. The GP-VAE using PointNet struggles to reconstruct even the

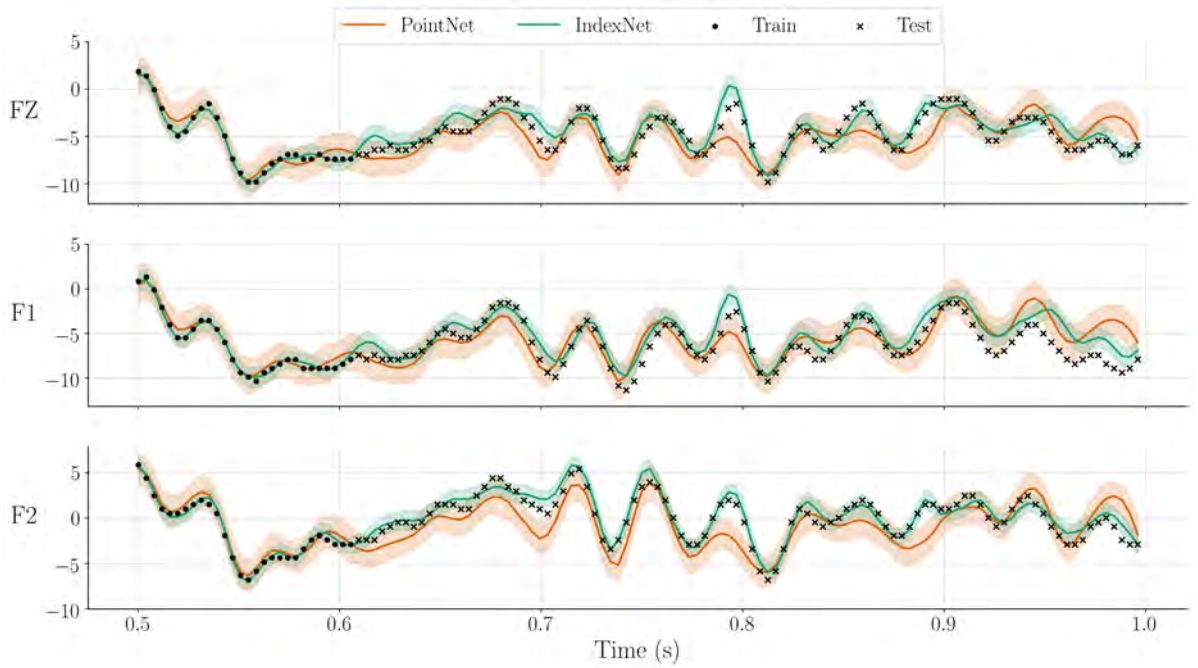


Fig. 5.3 A comparison between the posterior predictive distributions of two GP-VAE models using IndexNet (green) and PointNet (orange). The shaded region shows the 95% confidence interval for the predictive distributions.

observed data, resulting in large uncertainty estimates and poor NLL. The ineffectiveness of PointNet can be traced back to its assumption of smoothness across neighbouring dimensions. For the EEG dataset, the ordering of dimensions is not indicative of their position on the scalp, rendering the assumption invalid and a hindrance to the ability of the variational posterior to

[3] Although exact inference can be performed using a linear likelihood - see Teh et al. (2005) - the results reported here are better than those reported by Requeima et al. (2019) using exact inference in the linear model. This suggests two things: a) Requeima et al. (2019) found a poor local optimum and b) we do not lose much by performing approximate inference.

approximate the true posterior. The differences between the use of zero imputation, IndexNet and FactorNet is marginal. Although zero imputation is arguably less principled than the other two methods, the results indicate that, for the EEG dataset, the theoretical inadequacies of zero imputation do not translate to poor empirical performance.

To shed light on the properties of the GP-VAE, we consider repeating the experimental procedure laid out above with five, seven and nine latent dimensions. Figure 5.4 plots the average ELBO, SMSE and NLL achieved by GP-VAEs using zero imputation, IndexNet and FactorNet. The results using PointNet were significantly worse than those shown and so are omitted from the comparison. Provided the global optimum is found the average ELBO should increase

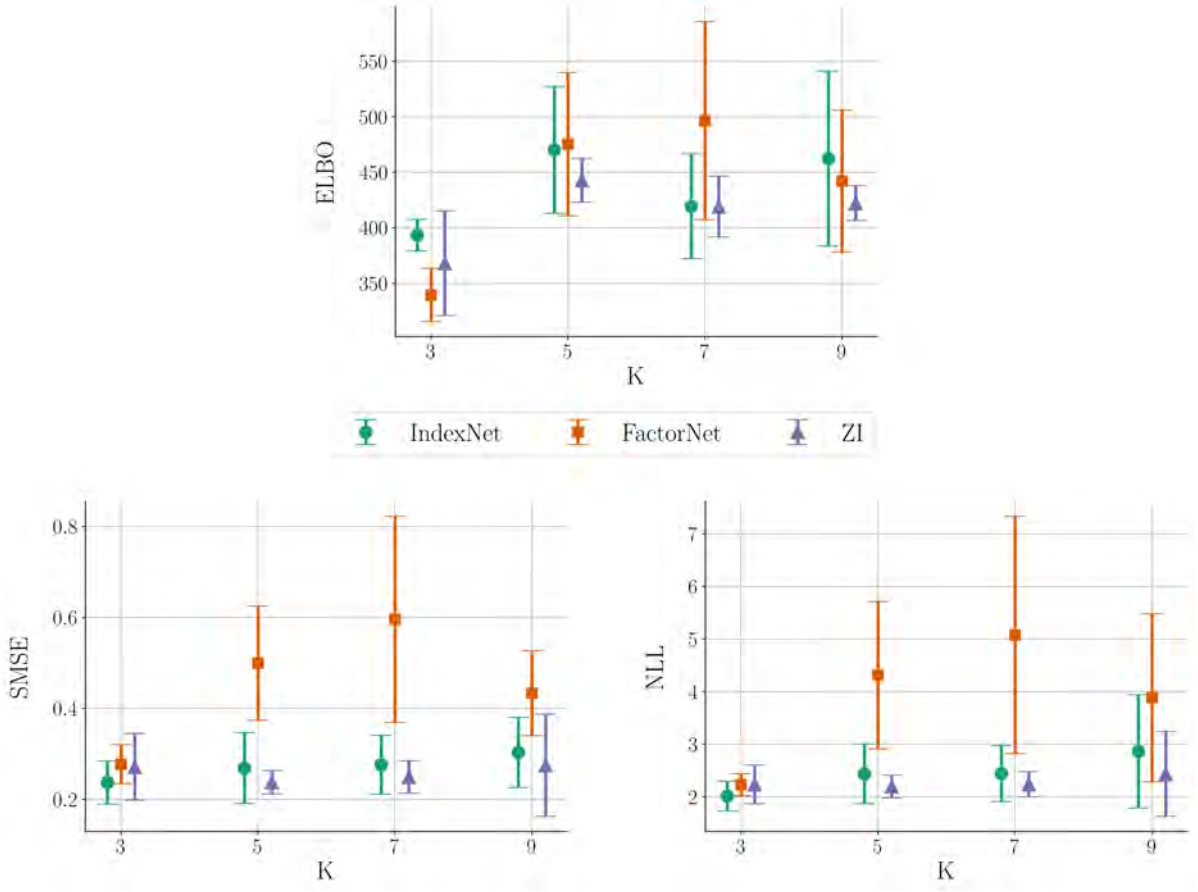


Fig. 5.4 Plots of the predictive performance of GP-VAE models as the number of latent dimensions varies. The error bars show the mean ± standard deviation averaged across 10 initialisations.

with number of latent dimensions for all GP-VAEs. Although this occurs when the number of latent dimensions is increased from three to five, the average ELBO for all GP-VAEs generally plateaus when the number of latent dimensions reaches five. This suggests that there is little

to gain from the use of more latent dimensions than observation dimensions. Considering that the computational complexity of the optimisation process increases linearly with the number of latent dimensions, using fewer latent dimensions than observed dimensions is advisable.

Whilst the average ELBO for all GP-VAEs remains comparable for all latent dimensions, the predictive performance for the GP-VAE using FactorNet becomes significantly worse than that of the GP-VAE using either IndexNet or zero imputation when more than three latent dimensions are used. Note that by increasing the number of latent dimensions, the GP-VAE is no longer forced to model dependencies between observed dimensions - the GP-VAE has the capacity to explain the data based on temporal correlations through the latent GPs alone. Strong performance on the EEG experiment hinges on the model's ability to model dependencies between observations, not temporal correlations. Indeed, a VAE using IndexNet and five latent dimensions achieves an average SMSE and NLL 0.201 ± 0.018 and 1.682 ± 0.161, yet an average ELBO of only -700.96 ± 19.92.

The procedure through which the GP-VAEs are trained does not necessitate the model to learn to reconstruct missing data from different patterns of partially observed data: maximisation of the ELBO corresponds to reconstructing only the data being conditioned on, not reconstructing data that is missing. This shortcoming can be addressed through a modification to the training objective. Specifically, rather than conditioning the approximate likelihood on all the observed data, $\mathbf{y}^o$, we condition the approximate likelihood on a subset of the observed data, $\tilde{\mathbf{y}^o}$:

$$q(\mathbf{f}) = \frac{1}{\mathcal{Z}_q(\theta, \phi)} p_{\theta_1}(\mathbf{f}) l_\phi(\mathbf{f}|\tilde{\mathbf{y}^o}). \tag{5.5}$$

The modified ELBO becomes

$$\mathcal{L}^o_{\text{ELBO}} = \sum_{n=1}^{N} \mathbb{E}_{q(\mathbf{f}_n)} \left[\frac{1}{\alpha} \log p_{\theta_1}(\mathbf{y}^o_n|\mathbf{f}_n) - \log l_\phi(\mathbf{f}_n|\tilde{\mathbf{y}^o}) \right] + \log \mathcal{Z}_q(\theta, \phi). \tag{5.6}$$

Note that likelihood of the observed data, $p_{\theta_1}(\mathbf{y}^o_n|\mathbf{f}_n)$, is retained. The modification encourages the GP-VAE to maximise the likelihood of data conditioned on partial observations.

We repeat the experimental procedure using the modified ELBO in (5.6) and construct $\tilde{\mathbf{y}^o}$ by removing a randomly selected 50% of values from $\mathbf{y}^o$ at each epoch. Figure 5.5 compares the predictive performance of the GP-VAEs as the number of latent dimensions is increased. As hypothesised, the use of the modified ELBO significantly improves the predictive performance of the GP-VAE using FactorNet, achieving an average SMSE of less than 0.3 for all latent dimensions. Although the average SMSE for IndexNet and zero imputation worsens to around 0.35, the average NLL for all GP-VAE models shows a significant improvement relative to the use of the original ELBO. The results suggest that the modified ELBO acts to regularise

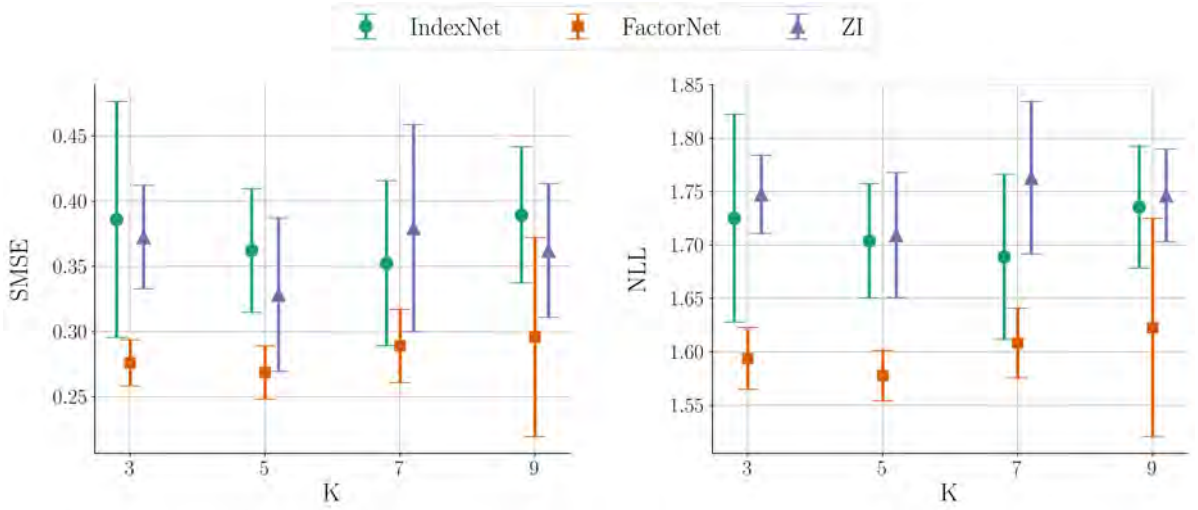


Fig. 5.5 The predictive performance of GP-VAE models trained using the modified ELBO in (5.6) as the number of latent dimensions K varies. The error bars show the mean ± standard deviation averaged across 10 initialisations.

the GP-VAE, deterring it from overfitting to the training data and, in turn, providing better predictive uncertainties.

Finally, although SE kernels were used for the latent GPs to permit a fair comparison with the results of Requeima et al. (2019), we found that by using a composition kernel formed by the addition of a periodic kernel to the SE kernel, a better fit to the data was obtained. Specifically, the GP-VAE using IndexNet, a three-dimensional latent space and composition kernels achieved an average SMSE and NLL of 0.188 ± 0.024 and 1.621 ± 0.102, respectively. The results represent a significant improvement upon the current state-of-the-art for multi-output GP models.

5.4 Jura Dataset

The Jura dataset is a geospatial dataset comprised of 359 measurements of the topsoil concentrations of several heavy metals - Cadmium, Copper, Lead, Cobalt, Cromium, Nickel and Zinc - collected from a 14.5km^2 region of the Swiss Jura (Goovaerts et al., 1997). Following the experimental procedure of others (Álvarez and Lawrence, 2011; Goovaerts et al., 1997; Requeima et al., 2019), a smaller dataset consisting of only three observations at each spatial location - Cadmium, Nickel and Zinc - is considered. The dataset is further divided into a training set consisting of Nickel and Zinc measurements for all 359 locations and Cadmium measurements for just 259 locations. Conditioned on the observed training set, the task is to predict the Cadmium measurements at the remaining 100 locations.

We use a two-dimensional latent space for all GP-VAE models with SE kernels with lengthscales and scales initialised to 1. Similar to the EEG data imputation task, using fewer latent dimensions than observed dimensions forces the model to share information across dimensions. Furthermore, this permits a fair comparison with other multi-output GP methods which also use two latent dimensions with SE kernels. For all DNNs except for those in IndexNet, we use two hidden layers of 20 units and ReLU activation functions. IndexNet uses DNNs with a single hidden layer of 20 units and a 20-dimensional intermediate representation. Following Goovaerts et al. (1997) and Lawrence (2004), the performance of each model is evaluated using the mean absolute error (MAE) averaged across 10 different initialisations. To account for the presence of poor local optima, the 10 different initialisations are identified from a body of 15 as those with the highest training set ELBO. For each initialisation the GP-VAE models are trained for 3000 epochs using a batch size of 100. We also report the average NLL for the GP-VAE models to indicate the quality of the posterior predictive distributions. Table 5.2 reports the performance of the GP-VAE models using zero imputation, IndexNet and FactorNet. Similar to the EEG dataset, their performance is compared to that of independent GPs, GPAR and the GP-VAE using IndexNet with linear likelihood (GP-VAE-LL).

Table 5.2 A comparison between multi-output GP models on the Jura data imputation task.

				GP-VAE		
Metric	IGP[†]	GPAR[†]	GP-VAE-LL	ZI	IndexNet	FactorNet
MAE	0.574	0.411	0.487 ± 0.008	0.420 ± 0.008	0.437± 0.020	**0.404 ± 0.005**
NLL	-	-	1.179 ± 0.041	1.131 ± 0.090	1.120 ± 0.082	**0.999 ± 0.062**

[†]Results taken directly from Requeima et al. (2019).

As observed with the EEG experiment, the results highlight the comparatively poor performance of independent GPs relative to multi-output GPs, demonstrating the importance of modelling correlations between metal concentrations. The GP-VAE using FactorNet achieves the best test set performance across all multi-output GPs. This holds true for models not included in Table 5.2, including Álvarez and Lawrence's (2011) convolved multi-output GP, Goovaerts et al.'s (1997) intrinsic coregionalisation model and Wilson et al.'s (2011) GP regression network which achieve average MAEs of 0.455, 0.461 and 0.453, respectively. The performance of the GP-VAE using zero imputation is also relatively strong, providing additional evidence that the use of zero imputation does not negatively impact the empirical performance of the GP-VAE. The GP-VAE using IndexNet achieves the worst test set performance out of all GP-VAEs with an average MAE of 0.437. Whilst this is substantially stronger than when using a linear likelihood function, it contrasts with its state-of-the-art performance on the EEG experiment in which the use of IndexNet resulted in the best test set performance and the use of FactorNet resulted in the worst. Such variability is undesirable and exposes a potential shortcoming of the GP-VAE.

Whilst the use of an inference network is unnecessary for either the EEG or Jura experiment, the state-of-the-art performance of the GP-VAE relative to other multi-output GPs demonstrates its ability to learn approximate posteriors of high quality. Nevertheless, it is instructive to consider what is lost through amortisation. To this end, we repeat the EEG and Jura experiments for the best performing GP-VAE models with the means and variances of the approximate likelihood freely optimisable. The model parameters are kept fixed to the optimum found by the amortised GP-VAE, such that any changes in the ELBO are due to improvements in the approximate posterior. Tables 5.3a and 5.3b report the results. As alluded to in Chapter 2,

Table 5.3 A comparison between the performance of the amortised GP-VAE and non-amortised GP-VAE (GP-VAE*).

(a) EEG

Metric	GP-VAE	GP-VAE*
SMSE	**0.238 ± 0.019**	0.251 ± 0.025
NLL	**2.012 ± 0.283**	2.127 ± 0.371
ELBO	393.5 ± 14.3	**428.4 ± 11.0**

(b) Jura

Metric	GP-VAE	GP-VAE*
MAE	**0.404 ± 0.005**	0.412 ± 0.005
NLL	**0.999 ± 0.062**	1.042 ± 0.060
ELBO	-991.9 ± 9.7	**-965.8 ± 2.6**

the use of amortisation results in a lower ELBO, indicating a worse approximation to the true posterior. However, this does not translate to poorer performance. For both the EEG and Jura experiments, the performance of the amortised GP-VAE is noticeably better than without amortisation. Importantly, the results demonstrate that the use of amortisation is not at the expense of predictive performance.

In the following experiments, we consider modelling datasets in which the use of existing multi-output GP models is unsuitable and the use of amortised VI is necessary.

5.5 Bouncing Ball Experiment

First introduced by Johnson et al. (2016) for evaluating the SVAE, and later considered by Lin et al. (2018) for evaluating the SIN, the bouncing ball experiment considers a sequence of one-dimensional images representing a ball bouncing under linear dynamics, as illustrated in Figure 5.6. The dataset consists of 80 12-dimensional image sequences each of length 50, with the task being to predict the trajectory of the ball given a prefix of a longer sequence. The image sequences are generated at random by uniformly sampling the starting position of the ball whilst keeping the bouncing frequency fixed.

To ensure a fair comparison with the SVAE and SIN, we adopt an identical architecture for the inference network and decoder. In particular, we use DNNs with two hidden layers of 50 units and hyperbolic tangent activation functions. Whilst both Johnson et al. and Lin et al. use eight-dimensional latent spaces, any image of the bouncing ball can be summarised by a

single value indicating its position in the one-dimensional image. Thus, we consider modelling the image sequences using a GP-VAE with a one-dimensional latent space and a periodic GP kernel to reflect our a priori knowledge of periodic dynamics. Figure 5.6 compares the posterior latent GP and mean of the posterior predictive distribution with the ground truth for a single image sequence. Observe that the ground truth is reconstructed with almost exact precision, owing in equal measure to

1. the ability of the GP prior to model the latent dynamics;
2. the flexibility of the likelihood function to map to the high-dimensional observations.

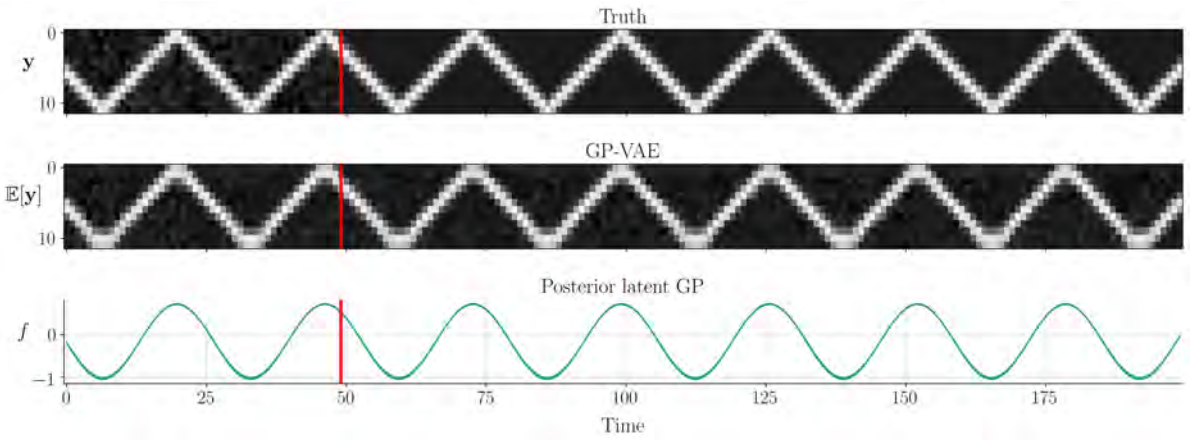


Fig. 5.6 A comparison between the mean of the GP-VAE's posterior predictive distribution (middle) and the ground truth (top) conditioned on noisy observations up to the red line. The latent approximate GP posterior is also shown (bottom).

Following Lin et al. (2018), we evaluate the τ-steps ahead predictive performance of the GP-VAE using the mean absolute error, defined as

$$\sum_{n=1}^{N_{\text{test}}} \sum_{t=1}^{T-\tau} \frac{1}{N_{\text{test}}(T-\tau)d} \left\| \mathbf{y}^*_{n,t+\tau} - \mathbb{E}_{q(y_{n,t+\tau}|y_{n,1:t})} \left[\mathbf{y}_{n,t+\tau} \right] \right\|_1 \tag{5.7}$$

where N_{test} is the number of test image sequences with T time steps and $\mathbf{y}^*_{n,t+\tau}$ denotes the noiseless observation at time step $t+\tau$. We use $N_{\text{test}} = 10$ and repeat the experiment 10 times to obtain a mean and standard deviation. Figure 5.7 compares the performance of the GP-VAE with the SVAE and SIN using LDS priors, alongside the benchmark performance of the regular LDS. Despite using just a single latent dimension, the GP-VAE significantly outperforms the other two models. These results demonstrate the GP-VAE's effectiveness in modelling high-dimensional data with low-dimensional latent dynamics.

To showcase the versatility of the GP-VAE, we extend the complexity of the original bouncing ball experiment to consider a sequence of 50-dimensional images representing two bouncing

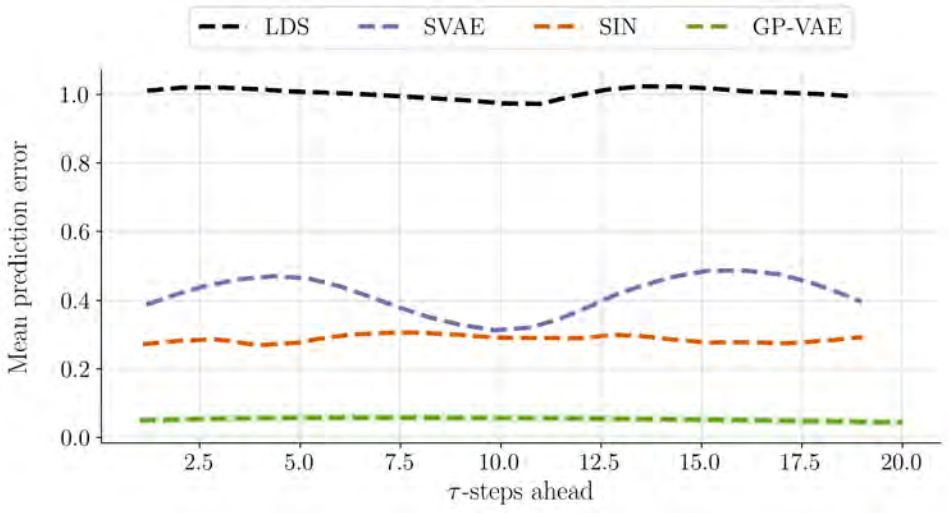


Fig. 5.7 A comparison between the τ-steps ahead predictive performance of the GP-VAE, SVAE, SIN and regular LDS. For the GP-VAE the mean $\pm$ standard deviation is shown. The results for the SVAE, SIN and regular LDS are taken directly from Lin et al. (2018).

balls: one under linear dynamics and the other under gravity. Furthermore, the images are corrupted by removing 50% of the pixels at random. As before, the dataset consists of 80 noisy image sequences, each of length 50, with the task being to predict the trajectory of the balls given a prefix of a longer sequence. We introduce a second latent dimension with a periodic kernel to model the latent dynamics of the second ball and the use of IndexNet to handle the partially observed data. To handle the increased number of observation dimensions, the number of hidden units is increased to 128. Figure 5.8 compares the posterior latent GPs and mean posterior predictive distribution with the ground truth for a single image sequence. Observe that the GP-VAE has 'disentangled' the dynamics of each bouncing ball, using a single latent dimension to model each. Similar to the simpler experiment, this enables the GP-VAE to recover the ground truth with impressive precision.

5.6 Weather Station Data

The Global Historical Climatology Network (GHCN) is a publicly available database consisting of monthly and daily climate summaries from over 100,000 weather stations situated across the globe[4]. Each climate summary comprises measurements of precipitation and temperature, alongside a multitude of other less frequently reported variables. Weather data such as this exhibits both complex spatio-temporal correlations as well as dependencies between observed variables, making it notoriously difficult to model. Furthermore, the dataset is extremely sparse and irregularly sampled, rendering models that place rigid assumptions on the structure of the data or pattern of missingness, such as the SVAE/SIN using LDS priors and GPAR, obsolete.

[4]The data can be found at https://www.ncdc.noaa.gov/ghcn-daily-description.

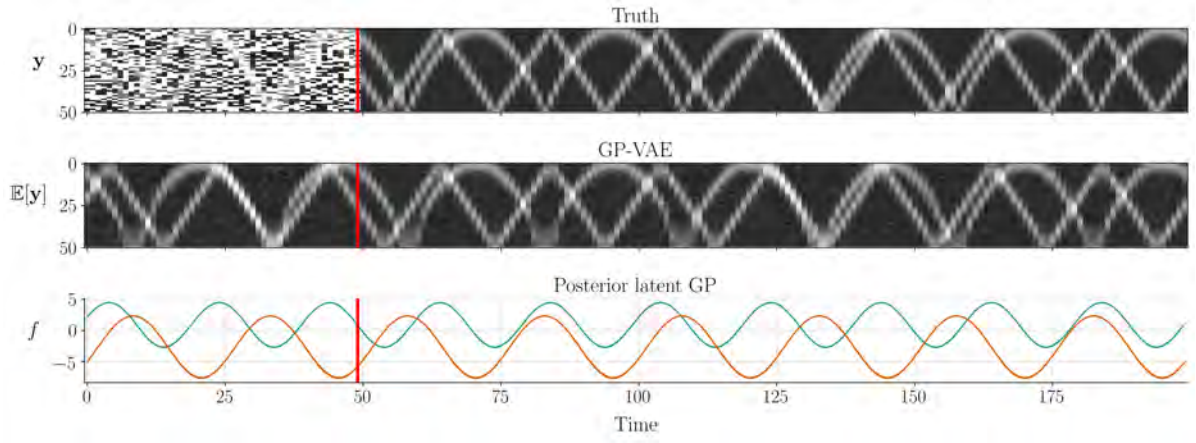


Fig. 5.8 An illustration of the mean of the GP-VAE's posterior predictive distribution (middle) and the ground truth (top) conditioned on noisy observations up to the red line. The latent approximate GP posterior is also shown.

5.6.1 Small Japanese Weather Experiment

We construct a subset of the data comprised of 731 daily climate reports from 156 Japanese weather stations throughout 1980 and 1981. Each report measures the daily precipitation (mm), maximum, minimum and average temperature (°C) as well as snow depth (mm), any pattern of which is potentially missing. The spatial location of each weather station is determined by its latitude, longitude and elevation above sea level (m). The rates of missingness in the dataset vary, with 6.3%, 14.0%, 18.9%, 47.3% and 93.2% of values missing for each of the five weather variables, respectively. The total number of data points present in the dataset is $156 \times 731 = 114036$, too many to be modelled by exact GPs. Instead, we group the data into days of three, with each group containing $156 \times 3 = 468$ data points distributed across 156 spatial locations and three temporal locations. We consider the task of predicting the average temperature at all stations conditioned on all other observations for that day together with all observations from the day before and after, as illustrated in Figure 5.9. Each model is trained on the 122 groups from 1980 and evaluated on the data from both 1980 and 1981. Not only does this assess the ability of the model to condition on observed data, but also its ability to generalise inference to unseen data.

Each model is trained using the data from a single group per update for 10 epochs, with the performance evaluated using the root mean squared error (RMSE) and negative log-likelihood (NLL) averaged across 10 independent runs with different initialisations. We compare the performance of the GP-VAE using IndexNet with that of the same model using a linear likelihood function, the VAE using IndexNet and independent GPs implemented using GPyTorch (Gardner et al., 2018). A naïve baseline using mean imputation is also provided. The IndexNet DNNs consist of two hidden layers of 50 units each with ReLU activation functions and a 50-dimensional intermediate representation. For the GP-VAE and VAE, we use a three-dimensional latent

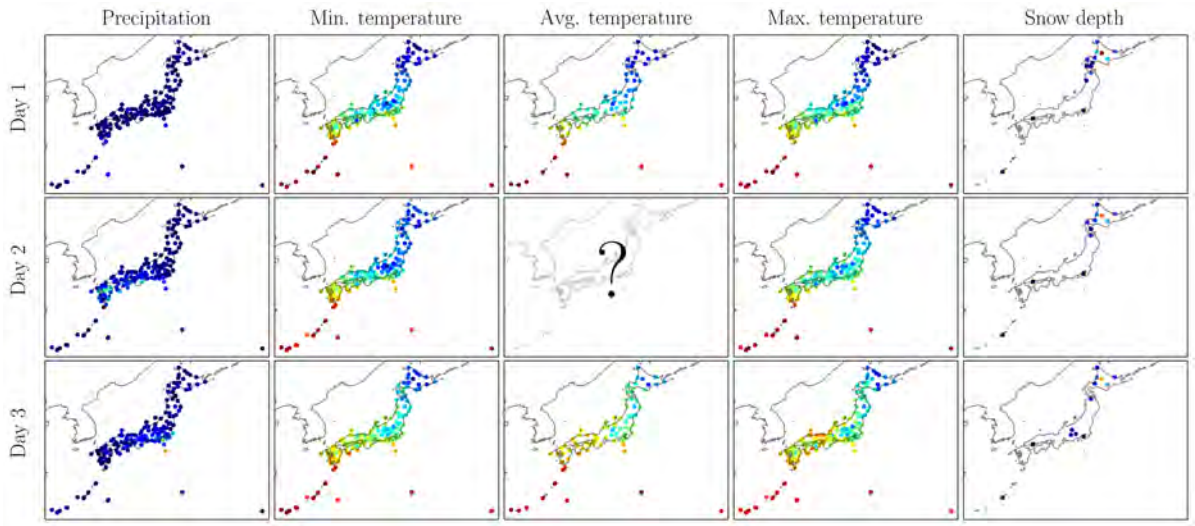


Fig. 5.9 An illustration of the small Japanese weather experiment.

space and decoder DNN consisting of two hidden layers of 20 units with ReLU activation functions. SE kernels are used for the GP-VAE and independent GPs with lengthscales and output scales initialised to 1. Tables 5.4a and 5.4b report the results and Figure 5.10 illustrates the mean of the posterior predictive distribution of the GP-VAE.

Table 5.4 The results for the small Japanese weather experiment.

(a) 1980

Metric	Baseline	IGP	VAE	GP-VAE-LL	GP-VAE
RMSE	9.191	2.179 ± 0.022	1.857 ± 0.081	1.678 ± 0.045	**1.536 ± 0.059**
NLL	-	2.800 ± 0.113	2.096 ± 0.056	2.081 ± 0.076	**1.924 ± 0.051**

(b) 1981

Metric	Baseline	IGP	VAE	GP-VAE-LL	GP-VAE
RMSE	9.660	2.118 ± 0.022	1.641 ± 0.112	1.510 ± 0.044	**1.502 ± 0.058**
NLL	-	2.679 ± 0.105	1.979 ± 0.070	1.911 ± 0.043	**1.906 ± 0.043**

All models significantly outperform the mean imputation baseline, which provides a poor estimate of the average daily temperature due to large seasonal and regional fluctuations. The vanilla VAE slightly outperforms independent GPs, suggesting that instantaneous dependencies between variables are more informative than the spatio-temporal dependencies of individual variables. The GP-VAE, however, is able to model both, achieving the best average RMSE and NLL on the 1980 dataset of 1.536 and 1.924, respectively. All models are able to generalise

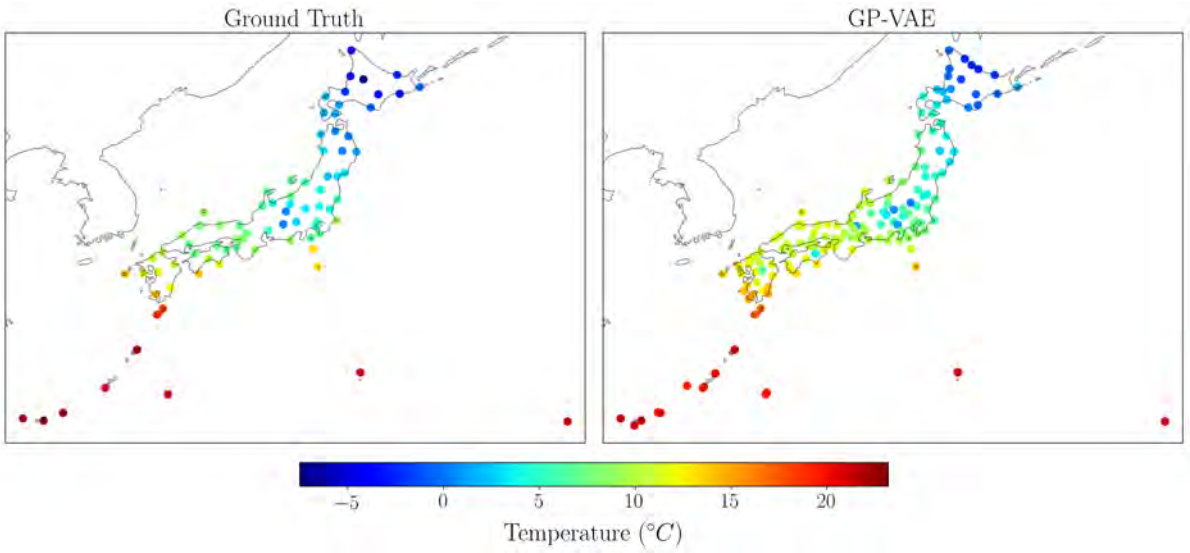


Fig. 5.10 A comparison between the ground truth average temperature readings for the 2nd January 1980 and the GP-VAE predictions. The GP-VAE predicts the average temperature reading for all weather stations, including those without ground truth values, conditioned on the measurements of all other weather variables and the measurements on the 1st and 3rd of January 1980.

inference to the unseen 1981 data, with the GP-VAE also achieving the best average RMSE and NLL of 1.502 and 1.906, respectively.

5.6.2 Large Japanese Weather Experiment

Rather than grouping the dataset into groups of three days, in this experiment we consider weekly groups consisting of $7 \times 156 = 1092$ data points each. Due to the computational complexity of exact GPs scaling cubically with the number of data points, the cost associated with learning and inference in this experiment using GP-VAEs is more than eight times that of the previous experiment. Instead, we turn towards using the sparse GP-VAE framework laid out in Chapter 3. We consider an alternative setup to the last, in which daily minimum, maximum and average temperature measurements are removed for all stations on any particular day with 10% probability. The task is to predict the missing values. Similar to the previous task, each model is trained on all the data from 1980 using a single group per update for 25 epochs and the performance evaluated on the data from 1980 and 1981 using RMSE and NLL averaged across 10 runs.

There is no existing framework for performing approximate inference in GP models conditioned on previously unobserved data, thus we cannot provide any comparison. Instead, we compare the performance of the sparse GP-VAE with that of the same model using a linear likelihood, a VAE and regular mean imputation. Both the sparse GP-VAEs and VAE employ IndexNet to handle partially observed data, with DNNs consisting of two hidden layers of 50 units and

ReLU activation functions, and a 50-dimensional intermediate representation. The decoder DNN uses two hidden layers of 20 units. For the sparse GP-VAE, we use a three-dimensional latent space and SE kernels with lengthscale and output scale initialised to 1. We implement the sparse GP-VAE using inducing points shared across each dimension and group, initialised using k-means clustering. The inducing point locations are treated as variational parameters to be optimised during training. Tables 5.5a and 5.5b report the results using 100 inducing points and an inference network that parameterises the nearest 20 inducing points.

Table 5.5 The results for the large Japanese weather experiment.

(a) 1980

Metric	Baseline	VAE	SGP-VAE-LL	SGP-VAE
RMSE	9.299	4.500 ± 0.260	3.520 ± 0.242	**3.259 ± 0.165**
NLL	-	5.862 ± 1.049	2.980 ± 0.127	**2.728 ± 0.087**

(b) 1981

Metric	Baseline	VAE	SGP-VAE-LL	SGP-VAE
RMSE	9.473	4.500 ± 0.155	3.449 ± 0.143	**3.020 ± 0.165**
NLL	-	5.862 ± 0.954	2.976 ± 0.109	**2.613 ± 0.088**

The sparse GP-VAE achieves the best average RMSE and NLL on both the 1980 and 1981 datasets, showcasing its ability to effectively condition the parameters of the approximate likelihood over inducing points on partially observed data. It should be emphasised that the total number of inducing points used for the entire dataset of 114036 data points is 100. Had the standard approach of Titsias (2009) been applied using independent sparse GPs, $2 \times 52 \times 3 \times 100 = 31200$ inducing points would be required, each with their own set of variational parameters. In a similar vein to the benefits of amortised VI versus mean-field VI, the sparse GP-VAE requires far fewer variational parameters than 'vanilla' sparse GPs and can generalise inference to unseen data. This offers substantial advantages when modelling large datasets such as this. Similar to previous tasks, the use of a linear likelihood function results in a noticeably worse predictive performance on the two datasets. Its inadequacy is exposed by the complexity of weather measurements, and the nonlinear dependencies between observations. Despite this, it still significantly outperforms the vanilla VAE which achieves the worst average RMSE and NLL of 4.500 and 5.862, respectively.

To shed light on how the number of inducing points, M, and the number of inducing points the inference network parameterises, T, affects the performance of the sparse GP-VAE, we repeated the experimental procedure for $M = 20$, 50 and 100 and evaluated the performance for $T = 2$, 5, 10 and 20. Figure 5.11 shows the variation in predictive performance for both

years. For $M = 100$, there is a general improvement in performance of the GP-VAE on both

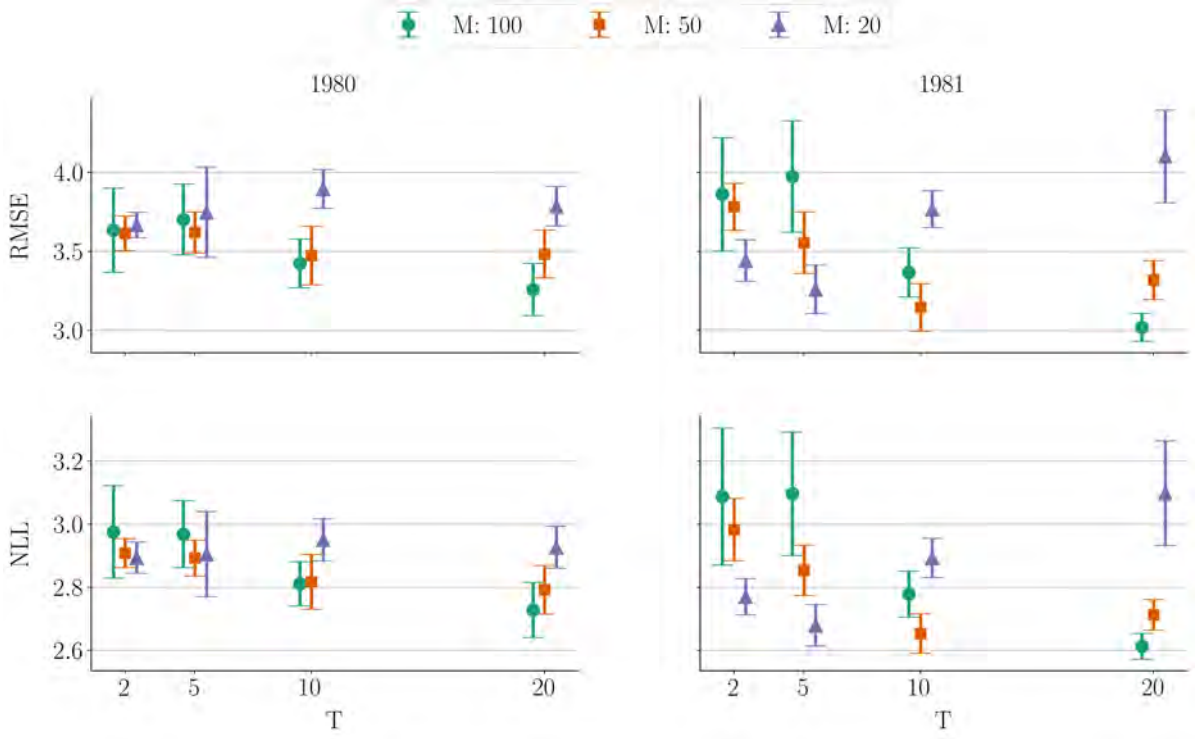


Fig. 5.11 The variation in predictive performance of the sparse GP-VAE on the large Japanese weather experiment as the number of inducing points, M, and inducing points parameterised by the inference network, T, vary. The plots show the mean ± standard deviation averaged across 10 independent initialisations.

datasets as T increases. Conversely, the performance of the GP-VAE with $M = 50$ worsens when T is increased from 10 to 20, and the performance of the GP-VAE with $M = 20$ worsens when T is increased beyond 5. This seems counter-intuitive - we would expect the accuracy of the approximate posterior to improve as T increases. Further, the inference network has a greater number of data points to train on for larger T. A second seemingly counter-intuitive phenomenon observed in Figure 5.11 is that the predictive performance is better using $M = 20$ than using either $M = 50$ or $M = 100$ when $T = 2$ is used. Similarly, the GP-VAE using $M = 100$ only outperforms that using $M = 50$ when $T = 20$.

Note that for a D-dimensional input space, the volume enclosed by the nearest T inducing points is, on average, proportional to $\frac{1}{M^D}$. This fact yields an explanation for both phenomena:

1. when the dependency between inputs is small, the approximate posterior does better assuming it to be 0 than trying to learn it. For small M, the value of T for which learning the dependency becomes a hindrance is also small. Thus, the predictive performance of the GP-VAE using $M = 20$ tails off first, followed by the GP-VAE using $M = 50$;

2. for large M and small T, the region of input space affected by each observation is too small for the approximate posterior to closely approximate the true posterior. Reducing M increases this volume, resulting in better predictive performance.

Thus, for any number of M inducing points, T must be chosen to strike a balance between these two opposing effects.

6 | Conclusion

The purpose of this thesis was to advance spatio-temporal dataset modelling through the establishment of a framework for the amalgamation of Gaussian processes and variational autoencoders. We defined a probabilistic model capable of explaining complex, multi-dimensional observations and the dependencies between them. Inference in the model is intractable; necessitating the use of variational inference. Particular attention was paid to the preservation of structure in the approximate posterior, ensuring the effectiveness of the model was realised. A unifying relationship between the developments made in this thesis and the work of others, in particular the family of multi-output Gaussian processes, was established. Finally, we conducted a rigorous empirical evaluation of the model on a variety of experiments involving datasets with differing characteristics.

In carrying out this research, we made a number of important contributions:

GP-VAE: we introduced a novel family of VAEs for modelling spatio-temporal data - the GP-VAE - characterised by the use of a GP prior over latent space and a structured approximate posterior. The theoretical framework necessary for performing VI was underpinned by the development of a number of Monte Carlo estimators, each of which was empirically evaluated. Crucially, we found that the strongest performing estimators were those whose performance hinged upon the quality of the approximate posterior, providing evidence in support of its ability to accurately approximate the true posterior.

Partial inference networks: we extended the suite of existing partial inference networks to include IndexNet, which was shown to offer distinct advantages, both theoretically and empirically, over the PointNet approach of Ma et al. (2019) and the product of Gaussians approach of Vedantam et al. (2017). Used together with the GP-VAE, we demonstrated state-of-the-art performance relative to other multi-output GPs and structured VAE models.

Sparse GP-VAE: to address the computational burden associated with inference in exact GPs, we extended the GP-VAE framework to accommodate sparse approximations. Marking a deviation from the existing and widely used approach of Titsias (2009), we introduced the

use of an inference network for parameterising the sparse GP approximation. Akin to the contributions and benefits of amortised VI relative to mean-field VI, our approach is the first to enable inference in sparse GPs on previously unobserved data with no additional training. We demonstrated the efficacy of the approach on a large, extremely sparse spatio-temporally distributed weather dataset and conducted a thorough investigation into its performance.

6.1 Future Work

We anticipate that the family of spatio-temporal VAEs developed in this thesis will advance the deployment of DLVMs on data rich with structure and dependencies. Nonetheless, we firmly believe that our developments have only scratched the surface of what the GP-VAE can achieve. We envision several research directions to be particularly promising:

State-space GP models: it is possible to reformulate temporal GPs as state-space models, which reduces the computational complexity from $\mathcal{O}\left(N^3\right)$ to $\mathcal{O}\left(N\right)$ using Kalman filtering and smoothing (Hartikainen and Särkkä, 2010). More recently, this has been extended to the spatio-temporal setting using infinite-dimensional extensions of the filtering and smoothing algorithms (Solin, 2016). In theory, these ideas could be used to significantly reduce the computational complexity associated with learning and inference in the GP-VAE. This will undoubtedly be challenging, not only due to the notorious difficulty of implementing the state-space equivalent GP: for non-Gaussian likelihoods, such as those employed in this thesis, the Kalman equations cannot be implemented exactly, demanding the use of approximate inference. This draws in new challenges which, generally speaking, diminish the advantages of translating the GP into state-space form (Chang et al., 2020). Addressing these limitations is necessary before application to the GP-VAE.

Streaming multi-output sparse GPs: in Chapter 3 we discussed the advantages of the sparse GP-VAE in comparison to existing sparse GP methods. Most notably, there is no reason why inducing points cannot be added, moved around or removed as desired - the purpose and structure of the inference network is unaffected. The ability to incrementally change the complexity of the sparse GP posterior in this manner has been previously unrealisable, and opens the door to a realm of avenues to explore with the sparse GP-VAE. One exciting application is to use the sparse GP-VAE to model high-dimensional data streamed through time. In theory, inducing points can be introduced sequentially to model the regions of newly observed data without any additional training.

Mixture of latent priors: whilst this thesis concerned itself with the use of GP priors in DLVMs, there may be settings in which it is advantageous to use a mixture of GP priors and standard normal priors over the latent dimensions - the purpose of the latent

dimensions with standard normal priors being to model information specific to each observation, such as the local geographical features of weather stations. Of course, this is speculative and somewhat idealistic; preliminary experiments on the Jura and Japanese weather datasets found no improvements in predictive performance when used. Whether this result is universal or dataset specific is unknown and demands further experimentation.

A | Mathematical Derivations

A.1 Posterior Gaussian Process

For the sake of notational convenience, we shall drop the subscript k from this derivation. Recall the approximate posterior over the latent function values, $\mathbf{f}$:

$$q(\mathbf{f}) = \frac{1}{\mathcal{Z}_q} \underbrace{\mathcal{N}\left(\mathbf{f}; \mathbf{0}, \mathbf{K_{ff}}\right)}_{p_{\theta_1}(\mathbf{f})} \underbrace{\mathcal{N}\left(\mathbf{f}_k; \boldsymbol{\mu}_\phi, \boldsymbol{\Sigma}_\phi\right)}_{l_\phi(\mathbf{f}|\mathbf{y})}. \tag{A.1}$$

$q(\mathbf{f})$ is a Gaussian with mean and covariance given by

$$\hat{\boldsymbol{\mu}} = \mathbf{K_{ff}}\left(\mathbf{K_{ff}} + \boldsymbol{\Sigma}_\phi\right)^{-1} \boldsymbol{\mu}_\phi \tag{A.2}$$

$$\hat{\boldsymbol{\Sigma}} = \mathbf{K_{ff}}\left(\mathbf{K_{ff}} + \boldsymbol{\Sigma}_\phi\right)^{-1} \boldsymbol{\Sigma}_\phi = \left(\mathbf{K_{ff}^{-1}} + \boldsymbol{\Sigma}_\phi^{-1}\right)^{-1}. \tag{A.3}$$

The approximate posterior over some latent function value f_* is obtained by marginalisation of the joint distribution:

$$\begin{aligned} q(f_*) &= \int p_{\theta_1}(f_*|\mathbf{f}) q(\mathbf{f}) d\mathbf{f} \\ &= \int \mathcal{N}\left(f_*; k_{f_*\mathbf{f}}\mathbf{K_{ff}^{-1}}\mathbf{f}, k_{f_*f_*} - k_{f_*\mathbf{f}}\mathbf{K_{ff}^{-1}}k_{\mathbf{f}f_*}\right) \mathcal{N}\left(\mathbf{f}; \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}\right) d\mathbf{f} \\ &= \mathcal{N}\left(f_*; k_{f_*\mathbf{f}}\mathbf{K_{ff}^{-1}}\hat{\boldsymbol{\mu}}, k_{f_*f_*} - k_{f_*\mathbf{f}}\left(\mathbf{K_{ff}^{-1}} - \mathbf{K_{ff}^{-1}}\hat{\boldsymbol{\Sigma}}\mathbf{K_{ff}^{-1}}\right)^{-1} k_{\mathbf{f}f_*}\right). \end{aligned} \tag{A.4}$$

Substituting (A.2) into the expression for the mean above gives rise to the GP mean function:

$$\hat{m}\left(\mathbf{x}\right) = k_{\mathbf{f}\mathbf{f}}\left(\mathbf{K_{ff}} + \boldsymbol{\Sigma}_\phi\right)^{-1} \boldsymbol{\mu}_\phi. \tag{A.5}$$

The covariance function is a little trickier. First, expanding the matrix $\mathbf{K_{ff}^{-1}} - \mathbf{K_{ff}^{-1}}\hat{\boldsymbol{\Sigma}}\mathbf{K_{ff}^{-1}}$ using (A.3) gives

$$\mathbf{K_{ff}^{-1}} - \mathbf{K_{ff}^{-1}}\hat{\boldsymbol{\Sigma}}\mathbf{K_{ff}^{-1}} = \mathbf{K_{ff}^{-1}} - \mathbf{K_{ff}^{-1}}\left(\mathbf{K_{ff}^{-1}} + \boldsymbol{\Sigma}_\phi^{-1}\right)^{-1}\mathbf{K_{ff}^{-1}}. \tag{A.6}$$

Now we can apply the matrix inversion lemma to the right hand side of (A.6) to give

$$\mathbf{K}_{\mathbf{ff}}^{-1} - \mathbf{K}_{\mathbf{ff}}^{-1}\left(\mathbf{K}_{\mathbf{ff}}^{-1} + \boldsymbol{\Sigma}_{\phi}^{-1}\right)^{-1}\mathbf{K}_{\mathbf{ff}}^{-1} = \left(\mathbf{K}_{\mathbf{ff}} + \boldsymbol{\Sigma}_{\phi}\right)^{-1}. \tag{A.7}$$

Substituting this into the expression for the covariance gives rise to the GP covariance function:

$$\hat{k}\left(\mathbf{x}, \mathbf{x}'\right) = k_{ff'} - k_{f\mathbf{f}}\left(\mathbf{K}_{\mathbf{ff}} + \boldsymbol{\Sigma}_{\phi}\right)^{-1} k_{\mathbf{f}f}. \tag{A.8}$$

A.2 Expected Gradient of the Approximate Likelihood

The expected gradient of the approximate likelihood, $\mathbb{E}_{q(\mathbf{f})}\left[\nabla_{(\cdot)} \log l_{\phi}(\mathbf{f}|\mathbf{y})\right]$, can be evaluated analytically as follows. First, we make the expansion

$$\begin{aligned}\mathbb{E}_{q(\mathbf{f})}\left[\nabla_{(\cdot)} \log l_{\phi}(\mathbf{f}|\mathbf{y})\right] &= \sum_{n=1}^{N}\sum_{k=1}^{K}\mathbb{E}_{q(f_{nk})}\left[\nabla_{(\cdot)} \log l_{\phi}(f_{nk}|\mathbf{y}_n)\right] \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K}\mathbb{E}_{q(f_{nk})}\left[\nabla_{(\cdot)}\left(-\log \sigma_{\phi,k}(\mathbf{y}_n) - \frac{1}{2\sigma_{\phi,k}^2(\mathbf{y}_n)}\left(f_{nk} - \mu_{\phi,k}(\mathbf{y}_n)\right)^2\right)\right].\end{aligned} \tag{A.9}$$

Application of the chain rule gives

$$\begin{aligned}\mathbb{E}_{q(f_{nk})}\left[\nabla_{(\cdot)} \log l_{\phi}(f_{nk}|\mathbf{y}_n)\right] = &-\nabla_{(\cdot)} \log \sigma_{\phi,k}(\mathbf{y}_n) \\ &- \mathbb{E}_{q(f_{nk})}\left[\left(f_{nk} - \mu_{\phi,k}(\mathbf{y}_n)\right)^2\right]\nabla_{(\cdot)}\left(\frac{1}{2\sigma_{\phi,k}^2(\mathbf{y}_n)}\right) \\ &+ \frac{1}{2\sigma_{\phi,k}^2(\mathbf{y}_n)}\mathbb{E}_{q(f_{nk})}\left[f_{nk} - \mu_{\phi,k}(\mathbf{y}_n)\right]\nabla_{(\cdot)}\mu_{\phi,k}(\mathbf{y}_n).\end{aligned} \tag{A.10}$$

We can expand the expectation $\mathbb{E}_{q(f_{nk})}\left[\left(f_{nk} - \mu_{\phi,k}(\mathbf{y}_n)\right)^2\right]$ as

$$\begin{aligned}\mathbb{E}_{q(f_{nk})}\left[\left(f_{nk} - \mu_{\phi,k}(\mathbf{y}_n)\right)^2\right] &= \mathbb{E}_{q(f_{nk})}\left[\left(f_{nk} - \hat{\mu}_{k,n}\right)^2\right] + \left(\hat{\mu}_{k,n} - \mu_{\phi,k}(\mathbf{y}_n)\right)^2 \\ &= \left[\hat{\boldsymbol{\Sigma}}_k\right]_{nn} + \left(\hat{\mu}_{k,n} - \mu_{\phi,k}(\mathbf{y}_n)\right)^2\end{aligned} \tag{A.11}$$

giving

$$\begin{aligned}\mathbb{E}_{q(f_{nk})}\left[\nabla_{(\cdot)} \log l_{\phi}(f_{nk}|\mathbf{y}_n)\right] = &-\nabla_{(\cdot)} \log \sigma_{\phi,k}(\mathbf{y}_n) - \\ &\left(\left[\hat{\boldsymbol{\Sigma}}_k\right]_{nn} + \left(\hat{\mu}_{k,n} - \mu_{\phi,k}(\mathbf{y}_n)\right)^2\right)\nabla_{(\cdot)}\left(\frac{1}{2\sigma_{\phi,k}^2(\mathbf{y}_n)}\right) \\ &+ \frac{1}{2\sigma_{\phi,k}^2(\mathbf{y}_n)}\left(\hat{\mu}_{k,n} - \mu_{\phi,k}(\mathbf{y}_n)\right)\nabla_{(\cdot)}\mu_{\phi,k}(\mathbf{y}_n).\end{aligned} \tag{A.12}$$

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, page 265–283, USA, 2016. USENIX Association.

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

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

Gowtham Atluri, Anuj Karpatne, and Vipin Kumar. Spatio-temporal data mining: A survey of problems and methods. *ACM Computing Surveys (CSUR)*, 51(4):1–41, 2018.

Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Edwin V Bonilla, Kian M Chai, and Christopher Williams. Multi-task Gaussian process prediction. In *Advances in Neural Information Processing Systems*, pages 153–160, 2008.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. Generating sentences from a continuous space. In Yoav Goldberg and Stefan Riezler, editors, *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, 2016.

Thang Duc Bui. *Efficient Deterministic Approximate Bayesian Inference for Gaussian Process models*. PhD thesis, University of Cambridge, 2018.

M Yu Byron, John P Cunningham, Gopal Santhanam, Stephen I Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. In *Advances in Neural Information Processing Systems*, pages 1881–1888, 2009.

Alex Campbell and Pietro Liò. tvGP-VAE: Tensor-variate Gaussian process prior variational autoencoder. *arXiv preprint arXiv:2006.04788*, 2020.

Francesco Paolo Casale, Adrian Dalca, Luca Saglietti, Jennifer Listgarten, and Nicolo Fusi. Gaussian process prior variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 10369–10380, 2018.

Paul E. Chang, William J. Wilkinson, Mohammad Emtiyaz Khan, and Arno Solin. Fast variational learning in state-space Gaussian process models, 2020.

Chris Cremer, Xuechen Li, and David Duvenaud. Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning*, pages 1078–1086, 2018.

Lehel Csató and Manfred Opper. Sparse on-line Gaussian processes. *Neural Computation*, 14 (3):641–668, 2002.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

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

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222, 1987.

David Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014.

Harrison Edwards and Amos Storkey. Towards a neural statistician. *arXiv preprint arXiv:1606.02185*, 2016.

Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. GP-VAE: Deep probabilistic time series imputation. In *International Conference on Artificial Intelligence and Statistics*, pages 1651–1661, 2020.

Jacob R. Gardner, Geoff Pleiss, David Bindel, Kilian Q. Weinberger, and Andrew Gordon Wilson. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 7587–7597, 2018.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Jimenez Rezende, and S. M. Ali Eslami. Conditional neural processes. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 1690–1699, 2018.

Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2013.

Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, pages 721–741, 1984.

Samuel Gershman and Noah Goodman. Amortized inference in probabilistic reasoning. In *Proceedings of the annual meeting of the cognitive science society*, volume 36, 2014.

Zoubin Ghahramani. Bayesian non-parametrics and the probabilistic approach to modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984), 2013.

Zoubin Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521: 452–9, 05 2015.

Pierre Goovaerts et al. *Geostatistics for natural resources evaluation*. Oxford University Press on Demand, 1997.

Geoffrey Strizaker Grimmett et al. *Probability and random processes*. Oxford university press, 2020.

Jouni Hartikainen and Simo Särkkä. Kalman filtering and smoothing solutions to temporal Gaussian process regression models. In *2010 IEEE international workshop on machine learning for signal processing*, pages 379–384. IEEE, 2010.

W Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1), 1970.

James Hensman, Nicolò Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, page 282–290, 2013.

James Hensman, Alexander G. de G. Matthews, and Zoubin Ghahramani. Scalable variational gaussian process classification. In Guy Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, San Diego, California, USA, May 9-12, 2015*, volume 38 of *JMLR Workshop and Conference Proceedings*. JMLR, 2015.

José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *International Conference on Machine Learning*, pages 1861–1869, 2015.

Geoffrey E Hinton. Learning multiple layers of representation. *Trends in cognitive sciences*, 11 (10):428–434, 2007.

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.

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

Matthew J Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in Neural Information Processing Systems*, pages 2946–2954, 2016.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Michael Irwin Jordan. *Learning in graphical models*, volume 89. Springer Science & Business Media, 1998.

Henry J Kelley. Gradient theory of optimal flight paths. *Ars Journal*, 30(10):947–954, 1960.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *2nd International Conference on Learning Representations*, 2014.

Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M Blei. Automatic differentiation variational inference. *The Journal of Machine Learning Research*, 18(1):430–474, 2017.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Malte Kuss and Carl Edward Rasmussen. Assessing approximate inference for binary Gaussian process classification. *Journal of Machine Learning Research*, 6(Oct):1679–1704, 2005.

Neil D Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in Neural Information Processing Systems*, pages 329–336, 2004.

Neil D. Lawrence and Andrew J. Moore. Hierarchical Gaussian process latent variable models. In *Proceedings of the 24th International Conference on Machine Learning*, page 481–488, New York, NY, USA, 2007. Association for Computing Machinery.

Yingzhen Li and Richard E Turner. Rényi divergence variational inference. In *Advances in Neural Information Processing Systems*, pages 1073–1081, 2016.

Yingzhen Li, José Miguel Hernández-Lobato, and Richard E Turner. Stochastic expectation propagation. In *Advances in Neural Information Processing Systems*, pages 2323–2331, 2015.

Yinzheng Li. Topics in approximate inference. 2017.

Wu Lin, Nicolas Hubacher, and Mohammad Emtiyaz Khan. Variational message passing with structured inference networks. In *International Conference on Learning Representations*, 2018.

Chao Ma, Sebastian Tschiatschek, Konstantina Palla, José Miguel Hernández-Lobato, Sebastian Nowozin, and Cheng Zhang. EDDI: efficient dynamic discovery of high-value information with partial VAE. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 4234–4243, 2019.

David JC MacKay. *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology, 1992.

David JC MacKay. Introduction to Gaussian processes. *NATO ASI Series F Computer and Systems Sciences*, 168:133–166, 1998.

David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Alexander G de G Matthews, James Hensman, Richard Turner, and Zoubin Ghahramani. On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. *Journal of Machine Learning Research*, 51:231–239, 2016.

Charles A Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 7:2651–2667, 2006.

Thomas P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2001.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte Carlo gradient estimation in machine learning. *arXiv preprint arXiv:1906.10652*, 2019.

Alfredo Nazabal, Pablo M Olmos, Zoubin Ghahramani, and Isabel Valera. Handling incomplete heterogeneous data using VAEs. *Pattern Recognition*, page 107501, 2020.

Radford M. Neal. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, 1995.

Manfred Opper. A Bayesian approach to on-line learning. *On-line learning in neural networks*, pages 363–378, 1998.

John W. Paisley, David M. Blei, and Michael I. Jordan. Variational Bayesian inference with stochastic search. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. 2017.

Michael Pearce. The Gaussian process prior VAE for interpretable latent dynamics from pixels. In *Symposium on Advances in Approximate Bayesian Inference*, pages 1–12, 2020.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.

Carl Edward Rasmussen and Zoubin Ghahramani. Occam's razor. In *Advances in Neural Information Processing Systems*, pages 294–300, 2001.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

James Requeima, William Tebbutt, Wessel Bruinsma, and Richard E Turner. The Gaussian process autoregressive regression model (GPAR). In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1860–1869, 2019.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning*, volume 32, pages 1278–1286, 2014.

Geoffrey Roeder, Yuhuai Wu, and David K Duvenaud. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In *Advances in Neural Information Processing Systems*, pages 6925–6934, 2017.

Sam Roweis and Zoubin Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11(2):305–345, 1999.

Rishit Sheth, Yuyang Wang, and Roni Khardon. Sparse variational inference for generalized GP models. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1302–1311, 07–09 Jul 2015.

Arno Solin. *Stochastic differential equation methods for spatio-temporal Gaussian process regression*. PhD thesis, Aalto University, 2016.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 3738–3746, 2016.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pages 1057–1063, 2000.

Yee Whye Teh, Matthias W. Seeger, and Michael I. Jordan. Semiparametric latent factor models. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, 2005.

Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

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

R. E. Turner and M. Sahani. Two problems with variational expectation maximisation for time-series models. In D. Barber, T. Cemgil, and S. Chiappa, editors, *Bayesian Time series models*, chapter 5, pages 109–130. Cambridge University Press, 2011.

Ramakrishna Vedantam, Ian Fischer, Jonathan Huang, and Kevin Murphy. Generative models of visually grounded imagination. *arXiv preprint arXiv:1705.10762*, 2017.

Aki Vehtari, Andrew Gelman, Tuomas Sivula, Pasi Jylänki, Dustin Tran, Swupnil Sahai, Paul Blomstedt, John P Cunningham, David Schiminovich, and Christian Robert. Expectation propagation as a way of life: A framework for Bayesian inference on partitioned data. *arXiv preprint arXiv:1412.4869*, 2014.

Edward Wagstaff, Fabian Fuchs, Martin Engelcke, Ingmar Posner, and Michael A. Osborne. On the limitations of representing functions on sets. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 6487–6494, 2019.

Martin J Wainwright and Michael Irwin Jordan. *Graphical models, exponential families, and variational inference*. Now Publishers Inc, 2008.

Andrew Gordon Wilson, David A Knowles, and Zoubin Ghahramani. Gaussian process regression networks. *stat*, 1050:19, 2011.

John Winn and Christopher M Bishop. Variational message passing. *Journal of Machine Learning Research*, 6(Apr):661–694, 2005.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in Neural Information Processing Systems*, pages 3391–3401, 2017.