University of Nottingham


School of Mathematical Sciences

# Mitigating Model Misspecification with Variational Bayesian Inference

**Ines Krissaane**

A thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy

March 2024

# Abstract

Learning dynamical systems from data is an important modelling problem in which one approximates the underlying equations of motion governing the evolution of some system. The conventional approach involves utilising a dynamical model, often derived from expert knowledge to accurately replicate the real-world data. This system often involves the inference of interpretable parameters so that model predictions align with observed data. When a chosen model fails to adequately represent the entire unknown dynamic, the ability to extract meaningful information from a fitted model can be challenging. This thesis investigates strategies addressing dynamical model misspecification within the framework of Bayesian inference. We delve into the limitations of standard Bayesian inference methods, specifically for parameter estimation, uncertainty quantification, and prediction accuracy. In our pursuit of a robust inferential approach, we assess the effectiveness of various contemporary methods such as generalised variational methods for dynamic modelling. Additionally, we introduce novel strategies to address model discrepancy, employing both Gaussian Processes and Approximate Bayesian Computation methods. This research aims to advance our understanding of Bayesian inference under model misspecification and offers practical guidance on constructing robust inferential approaches for more accurate and reliable results in continuous-time dynamic process.

# Acknowledgements

Words cannot express my gratitude to my four supervisors for their invaluable patience and guidance throughout this journey. Thanks to Professor Richard Wilkinson for his enriching and benevolent supervision. Thank you to Professor Gary Mirams, Doctor Edward Meeds, and Professor Jeremy Oakley for their participation in my thesis monitoring, their scrupulous proofreading of the manuscript, and their always judicious suggestions.
I am also grateful to my defense committee who generously provided knowledge and expertise. Additionally, I would like to express my gratitude to Microsoft Research for their generous support, which funded a significant portion of my research. My sincere thanks go to the Cardiac Modelling group led by Professor Gary Mirams and the University of Nottingham for the warm welcome and the privileged working conditions offered to me.
I extend my heartfelt thanks to all my friends and family, wherever they may be in the world, for their belief in me and their contribution to fostering a growth mindset.

Je tiens à exprimer ma profonde reconnaissance à mon petit frère Nour et à mon grand frère Mehdi pour leur soutien moral indéfectible. C'est avec une immense joie et le cœur ému que je dédie cette thèse à mes chers parents, Evelyne et Noureddine. Ils sont les fondations de qui je suis et de ce que je fais. Leur soutien indéfectible est une constante source d'encouragement, constituant ma principale motivation dans tout ce que j'entreprends dans la vie.

Truth, like water, takes the
shape of the vase containing it.

---

Ibn Khaldun

# Contents

# List of Tables

# List of Figures

# Abbreviations

**ABC** Approximate Bayesian Computation.

**AD** Automatic Differentiation.

**dGp** data-generating process.

**ELBO** Evidence Lower Bound.

**GBI** Generalised Bayes Inference.

**GP** Gaussian Process.

**GVI** Generalised Variational Inference.

**HM** History Matching.

**KLD** Kullback Leibler Divergence.

**MCMC** Markov Chain Monte Carlo.

**MLE** Maximum Likelihood Estimation.

**ODE** Ordinary Differential Equation.

**PDE** Partial Differential Equation.

**RBF** Radial Basis Function kernel.

**RFFs** Random Fourier Features.

**SDE** Stochastic Differential Equation.

**UQ** Uncertainty Quantification.

**VAE** Variational Auto Encoder.

**VI** Variational Inference.

# Chapter 1

# Introduction & Motivation

**Summary:** In this opening chapter, we outline the challenges and motivations that serve as the foundation of this thesis. We emphasise the importance of addressing misspecification in mechanistic models to ensure accurate parameter estimation and meaningful interpretation. Additionally, we introduce variational Bayesian approaches with the goal of establishing a robust inference framework in modelling.

## 1.1 Wrong models are the best we can do

### 1.1.1 Models are approximations of the truth.

The aphorism "All models are wrong, but some are useful" from the statistician George E.P. Box highlights the idea that we should always keep in mind the approximate nature of any model. In his book *Empirical Model-Building and Response Surfaces* (Box and Draper [1987]), he wrote "Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful". In other words, any modelling procedure should recognise the inherent bias due to a truncated representation of reality (Grünwald [2016]).

Whether you are working with a simple statistical model, such as a linear

regression model with Gaussian errors, or a computer simulation that solves complex mathematical equations, for example, a simulation of the Earth's climate, we are often faced with the problem that in some regimes, the model is misspecified. Misspecification typically arises in situations where modelling assumptions are inaccurate, stemming from limitations in our understanding of the modeled process. This occurs when the model fails to encompass all relevant factors, such as missing essential physics or omitting significant variables. Such a situation is most often detected when a model fails to replicate the data-generating process (dGp) in some statistical sense, which will be explored in this thesis. The disparities between the model and the true data-generating process referred to as the *model discrepancy* introduce bias in inference and compromise the accuracy of predictions. Engaging with a misspecified model can lead to suboptimal forecasts, an erroneous comprehension of underlying processes, and, in practical terms, may result in costly decision-making.

### 1.1.2 But models should be considered uncertain...

In the context of this thesis, we recognise the inherent approximation within our chosen model class and endeavor to enhance our inferences. It is therefore crucial to emphasise the necessity of considering uncertainties, often stemming from inaccuracies in the constitutive model parameters, along with other sources like measurement errors, inherent noise, and natural variability.

The aleatory uncertainty refers to the inherent uncertainty in a system that often cannot be fully eliminated but can be reduced through better understanding or better data. This type of uncertainty arises due to inherently random factors, such as measurement error or natural variation in a system (Mirams et al. [2016]). For example, consider the process of measuring the

weight of an object using a scale. The scale may have inherent variability in its readings due to imperfections in the measurement mechanism or random fluctuations in the environment, leading to measurement error.
In contrast, epistemic uncertainty, also known as knowledge uncertainty or model uncertainty, refers to the uncertainty that arises due to a lack of knowledge or understanding about a system or process. There are a variety of reasons why model misspecification can arise, ranging from a lack of understanding of the process, simplified assumptions and incomplete or biased data. An example of epistemic uncertainty is in climate modelling. There may be uncertainty about the future state of the climate due to incomplete knowledge about the underlying physical mechanisms, such as the interactions between the atmosphere, oceans, and land surface. We argue in this thesis that a standard Bayesian treatment of a misspecified model can lead to unreliable epistemic uncertainty estimates. This is why we position this thesis within the realm of Uncertainty Quantification (UQ), a critical field of science that describes the uncertainty in model parameters and the consequent uncertainty in model outputs. The assessment and mitigation of uncertainty's impact in scientific and engineering applications are beneficial for avoiding biased predictions, incorrect parameter estimates, and poor model generalisation (Kennedy and O'Hagan [2001]).

### 1.1.3 ... to make them useful and meaningful.

A meaningful model should be simple enough to be easily understood and interpreted, yet complex enough to capture the essential features of the system being studied. In this thesis, we emphasise the importance of adopting a discrepancy approach rather than fully discarding a model that, despite being incorrect in some sense, still offers valuable insights into reality through physically meaningful quantities. Specifically, we de-

| Models | Mechanistic | Statistical |
|---|---|---|
| Knowledge | Theory-Driven | Data-Driven |
| Explanatory | Causal | Descriptive |
| Objective | Understanding | Prediction |
| Realism | Explicit process | No underlying process |
| Assumptions | Many/Complex | Few/Simple |
| Parameters | Meaningful | Often Meaningless |
| Scaling outside the domain | Possible | Usually not |

Table 1.1: Contrasting Mechanistic and Statistical Models: A Simplified Perspective

lineate between two types of models commonly employed in scientific and engineering disciplines, as depicted in Table 1.1.

*Mechanistic models* are based on a set of explicit physical or mathematical equations that describe the underlying mechanisms of a system. They typically rely on detailed knowledge of the system's structure and behaviour and are designed to simulate system responses to changes in its inputs or environment. Mechanistic models are often used in fields such as physics, engineering, and chemistry to predict system behaviour under different conditions. Mechanistic modelling offers the possibility of learning information, making decisions, and generating predictions and novel hypotheses, thereby enhancing our comprehension of complex world systems. They are built from empirical data and expertise in the domain relying on complex relationships modeled with a set of assumptions. Complex natural mechanistic models often take the form of an Ordinary Differential Equation (ODE), a Stochastic Differential Equation (SDE), or a Partial Differential Equation (PDE) to describe the dynamics of the state of a system. These models rest on knowing the physical or biological processes that gave rise to the data.

*Statistical models*, on the other hand, is based on analysing patterns in data and making predictions. They do not typically rely on detailed knowledge of the underlying mechanisms of the system, but rather use statistical algo-

rithms and techniques to find relationships in the data (Schölkopf [2019]). Mechanistic models, particularly those with non-linear dynamics, are often difficult to analyse statistically, namely when system parameters need to be estimated. While mechanistic models have hypothesised relationships between the variables in the data set where the nature of the relationship is specified in terms of the underlying processes that are thought to have given rise to the data, statistical models have often hypothesised relationships between the variables in the data set, where the relationship seeks only to describe the data best. For misspecified mechanistic models, it is well known that if we fail to account for the model discrepancy in our inference (Lei et al. [2020]), our parameter estimates, instead of being physically meaningful quantities, will have their meaning intimately tied to the model used to estimate them (we end up estimating 'pseudo-true' values).

In reality, the rigidity of mechanistic models (Roberts [2021]) often leads to misspecification, where no parameter setting can perfectly replicate the data. On the contrary, statistical models and more largely machine learning can offer innovative methods to understand the intricacies of real-world dynamics.

## 1.2 Beyond Traditional Modelling with Variational Inference

### 1.2.1 Physics informed models

In this thesis, our main goal is to address the challenge of integrating the strengths of mechanistic biological models with statistical models. The integration of these two approaches offers a unique opportunity to utilise the mechanistic knowledge embedded in biological models alongside the capabilities of data-driven models. Mechanistic models often suffer from compu-

tational complexity and parameter uncertainty, while data-driven methods can lack interpretability (Matei et al. [2020]) and may struggle with incorporating mechanistic insights. By reconciling these two approaches, we want to develop hybrid models capable of providing accurate predictions and interpretable insights into complex biological phenomena (Tulleken [1993]). This integration also allows us to address scenarios where mechanistic models alone may be inadequate due to incomplete knowledge or computational constraints ([Vapnik, 1998, Gherman et al., 2022]). The thesis will use variational Bayesian approaches for misspecified mechanistic models of real-world systems. The dominant computational approach for numerically solving Bayesian inference problems is based on sampling methods such as Markov chain Monte Carlo. In contrast, variational methods use optimization methods to approximate Bayesian posterior distributions and provide a framework for estimating model parameters and handling uncertainty by leveraging Bayesian principles. These approaches offer flexibility and adaptability in dealing with complex modelling scenarios and can help overcome limitations associated with model misspecification.

### 1.2.2 An introduction to variational Bayes

In the Bayesian modelling framework, we have a probabilistic model describing a data-generating process through a joint distribution of latent variables and the data.

Let $x_{1:n}$ be observed data in a sample space $\mathcal{X}$ generated independently and identically from a distribution $g(x)$. The likelihood $\mathcal{L}(\theta|x)$ is directly related to the probability of observing data $x$ under a particular parameter $\theta$, $p(x \mid \theta) := p(x_{1:n} \mid \theta)$ where $\theta \in \Theta$ is a parameter vector with values in $\Theta \subseteq \mathbb{R}^d$. When we say a model is misspecified, we mean $g(x) \notin \{p(x \mid \theta) : \theta \in \Theta\}$.

Let $p(\theta)$ be our prior distribution summarising all the a-priori knowledge about the model parameters. Bayes' theorem allows us to update our prior beliefs to account for the evidence coming from the observed data, into a posterior distribution with:

$$\begin{aligned} p(\theta \mid x) &= \frac{p(x \mid \theta)p(\theta)}{\int p(x \mid \theta)p(\theta)d\theta}, \\ &\propto p(\theta) \prod_{i=1}^{n} p(x_i \mid \theta). \end{aligned} \tag{1.1}$$

Here the notation $\propto$ means proportional up to the normalising constant that is independent of the parameter $\theta$, and $p(x|\theta)$ is the conditional probability density of $x$ given $\theta$ called the likelihood. Each $x_i$ follows a normal distribution, and their joint distribution is a product since the $x_i$s are independent.

Variational approximations represent a family of deterministic methods for approximating the posterior distribution. The key idea behind the variational estimator is to cast the inference problem as an optimization task, where the goal is to find the parameters that minimise the divergence between the true posterior and the variational approximation. This allows for scalable and efficient inference, making variational estimators increasingly popular in Bayesian statistics, especially in cases where the true posterior is analytically intractable or computationally expensive to compute ([Blei et al., 2016, Csiszar, 1975, Donsker and Varadhan, 1976, Wang and Blei, 2019, Jordan et al., 1999, Wainwright and Jordan, 2007]).

Variational approximations aim to find the distribution $q(\theta) \in \mathcal{Q}$ that is close to the posterior $p(\theta|x)$ according to a divergence measure $\mathcal{D}(q\|p)$ : $\mathcal{Q} \times \mathcal{P} \rightarrow \mathbb{R}^+$, where $p$ and $q$ are generic density functions belonging to the sets $\mathcal{P}$ and $\mathcal{Q}$. The optimal approximate posterior $q^*(\theta)$ represents the best approximation to the true posterior distribution among the possible

set of candidates $\mathcal{Q}$ and solves the following optimization problem:

$$q^*(\theta) = \arg\min_{q \in \mathcal{Q}} \mathcal{D}(q(\theta) \| p(\theta \mid x)). \tag{1.2}$$

Note that the divergence is not necessarily symmetric. Variational methods are known for their computational efficiency compared to simulation-based methods like Markov chain Monte Carlo (MCMC). However, they rely on defining a family of approximating densities and finding the element that minimises a divergence measure with the target distribution. One well-known variational inference technique is variational Bayes (VB), which involves minimising the Kullback-Leibler divergence (Kullback and Leibler [1951]).

In the realm of probability theory, the Kullback-Leibler Divergence (KLD), denoted as KLD($f \| g$), emerges as a crucial measure quantifying the dissimilarity between probability densities characterised by functions $f(x)$ and $g(x)$. Originating from the work of Solomon Kullback and Richard Leibler, this concept serves as a fundamental tool in information theory and statistics.

**Definition 1.1** (The Kullback-Leibler Divergence )**.** *The KLD between probability densities g(x) and f(x) is given by*

$$KLD(f \| g) = \int f(x) \log \frac{f(x)}{g(x)} dx.$$

The KLD essentially gauges the amount of information lost when one probability density $g(x)$ is employed to estimate or approximate another $f(x)$. Notably, the KLD is characterised by its asymmetry, placing a large penalty on deviations of $g(x)$ from $f(x)$ in regions where $f(x)$ has higher density. In contrast, regions where the density of $g(x)$ is larger than $f(x)$ are not penalised as much.

In variational Bayes, the objective is to find the optimal approximation $q^*(\theta)$ within a specified set of distributions $\mathcal{Q}$ that minimises the KLD between $q(\theta)$ and the true posterior $p(\theta|x)$ :

$$q^*(\theta) = \underset{q \in \mathcal{Q}}{\arg\min}\ \mathrm{KLD}\left(q(\theta) \| p(\theta \mid x)\right), \tag{1.3}$$

The KLD quantifies the discrepancy between the variational approximation $q(\theta)$ and the true posterior $p(\theta|x)$. Minimising the discrepancy using the Kullback-Leibler Divergence guides the variational approximation to closely match the Bayesian posterior distribution, forming a fundamental method in approximate inference within Bayesian analysis. This paradigm requires a suitable family of approximating densities $\mathcal{Q}$ (Ormerod and Wand [2010]). Different assumptions about the space $\mathcal{Q}$ lead to various variational paradigms (cf. Chapter 3 for more details).

## 1.3 Structure of this thesis

This thesis proposes to apply variational-based approaches to address the challenges of estimating uncertainties in parameter inference, particularly in complex mechanistic and statistical models. We study the statistical robustness of these approaches when the models are misspecified due to discrepancies between the model assumptions and the actual system being modeled. A key contribution of this thesis is the introduction of a novel class of posterior belief distributions to address model discrepancy in the context of structural dynamical misspecification. The thesis also explores various strategies from a wide spectrum of scientific fields to enhance inference procedures and obtain more accurate and reliable estimates. The findings underscore the potential of combining generalised robust losses in Variational Inference (VI) with the model discrepancy in misspecified

models, paving the way for robust inference in dynamical models. By adopting a grey-box modelling strategy, this research provides a unified and interpretable framework that effectively captures the mechanistic and data-driven aspects of the underlying dynamics. This innovative approach emphasises the importance of addressing model misspecification, especially for dynamic models.

In Chapter 2, we present various types of misspecification and delve into the impact of misspecification on standard statistical tasks, such as inference of $\theta$, calibrated prediction, and decision tasks. We primarily focus on Bayesian and generalised Bayes approaches, exploring a variety of methods aimed at mitigating the effects of misspecification. Additionally, we clarify the desirable properties of statistical methods when dealing with misspecified models, as well as the limitations of what can be realistically achieved.

In Chapter 3, we adopt a variational approach within a variational autoencoder, which will appear as a natural choice to do inference with a dynamical model. This sophisticated methodology is employed to conduct inference for dynamic models, offering an alternative to Markov Chain Monte Carlo (MCMC) techniques. At a methodological level, this enables the integration of recent advancements in variational inference, particularly the use of automatic differentiation for differentiable ODEs to be applied to the parameter estimation task in mechanistic models in well-specified and misspecified scenarios.

In Chapter 4, we focus on inference using a non-standard inference approach and empirically explore how different approaches to dealing with discrepancies work with several misspecified mechanistic models. Particularly we compute generalised posteriors through a Generalised Variational

Inference (GVI) procedure with a wise choice of robust losses including UQ-derived losses. Taking a robust model-driven approach, we improve the accuracy and reliability of statistical inference in the presence of mechanistic model misspecification. However, we also highlight the limitations of certain Generalised Bayes methods when addressing these less-explored forms of misspecification.

Chapter 5 proposes a novel approach for modelling discrepancies in the context of ordinary differential equation (ODE) models. Our method incorporates Gaussian Process approximation through Random Fourier features and seamlessly integrates differentiable programming to propagate gradient information through ODE solvers. The method demonstrates the flexibility of incorporating domain knowledge through hybrid models that combine mechanistic ODEs and data-driven techniques, embracing both the known and unknown aspects of the system dynamics.

Finally, in Chapter 6, we present the conclusions for every chapter, together with various directions for future work.

# Chapter 2

# Robust Inference with Misspecified Models: Current Approaches and Efficacy

**Summary:** This chapter serves as an introduction to the concept of robustness in statistics within the context of model misspecification, which forms the overarching framework for the thesis. We explore methods for robust inference when the parameterized statistical models are misspecified relative to the data-generating process. In such cases, standard Bayesian inference can often be restricted because it heavily depends on the chosen prior and statistical model, which may be subject to varying degrees of misspecification. Particularly, we focus on addressing misspecification in mechanistic models to ensure accurate parameter estimation and meaningful interpretation. The chapter investigates existing strategies for addressing this issue and considers whether these methods are sufficient.

## 2.1 Robust Inference

In this thesis, robust inference denotes a statistical approach that is still reliable and reasonably efficient under small (or large) deviations of the data-generating process (dGp) from the assumed model. The goal is to ensure accurate results and valid statistical inference even in the presence of misspecification, outliers, or other sources of variability. Robust statistics (Arjovsky et al. [2017], Huber [2011]) seek methodologies that maintain reliability and reasonable efficiency, even when encountering minor deviations from the true model. While robustness may lead to less efficient parameter estimates, this trade-off is crucial in mechanistic physical models where the parameters are intended to be meaningful since they are often associated with real-world physical quantities or processes.

Various inference methods have been developed to tackle the challenges posed by model misspecification, that is when the true dGp lies outside the space of models that we intend to use for inference. A robust inferential procedure will ideally converge to the correct $\theta$ parameter (*Bayesian consistency*) even if the model is misspecified.

## 2.2 Varying degrees of misspecification

Models can be wrong to varying degrees, and the level of corruption depends on how much the model assumptions deviate from the actual dGp (cf. Figure 2.1). An instance of a misspecified statistical model arises when a dataset exhibiting a non-linear correlation between independent and dependent variables is analysed using a linear regression model. In such scenarios, the linear regression model fails to accurately represent the genuine relationship between the variables, often resulting in biased estimations of the regression coefficients ([Grünwald and van Ommen, 2014,


Figure 2.1: Forms of model misspecification with varying degrees of misalignment between the green dGp unknown and the black fitted model.

Walker, 2013]).

We present simple examples to introduce and motivate the need for robustness when the model is misspecified. In Example 1, we first consider the misspecified linear model, where the true dGp has some more complex form.

**Example 1** (Bayesian Linear Regression)**.**

*In the standard linear regression model, the relationship between the predictor variable $x_i$ and the response variable $y_i$ is given by:*

$$y_i = \theta x_i + \epsilon_i \tag{2.1}$$

*where $\epsilon$ is i.i.d (independent and identically distributed) errors following a normal distribution with mean 0 and variance $\sigma^2$, and $\theta$ represents the model parameter.*

*Suppose the dGp model is the nonlinear model where we leave the parameter $a$ as a free parameter we can modify to control the degree of nonlinearity. The relationship between the predictor variable $x_i$ and the response variable $y_i$ in this model is given by:*

$$y_i = \theta \frac{x_i}{1 + \frac{x_i}{a}} + \epsilon_i \tag{2.2}$$

*Similarly to the linear regression model, $\epsilon_i$ is a random error term with a normal distribution having mean 0 and variance $\sigma^2$. The nonlinearity in*

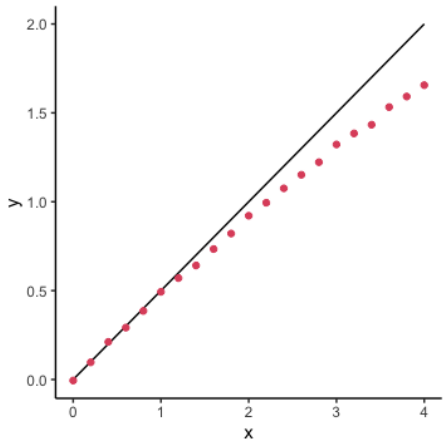



Figure 2.2: Red points show $n = 20$ observations from the dGp process Equation 2.2 with ($\theta = 0.5$, $\epsilon \sim \mathcal{N}(0, 0.01^2), a = 20$). The solid black line represents the linear model ($y = \theta x$) with $\theta = 0.5$.

*this model arises from the denominator* $1 + \frac{x_i}{a}$. *When* $a \to \infty$, *we retrieve the linear regression model. For small values of* $a$, *the linear regression model is highly misspecified.*

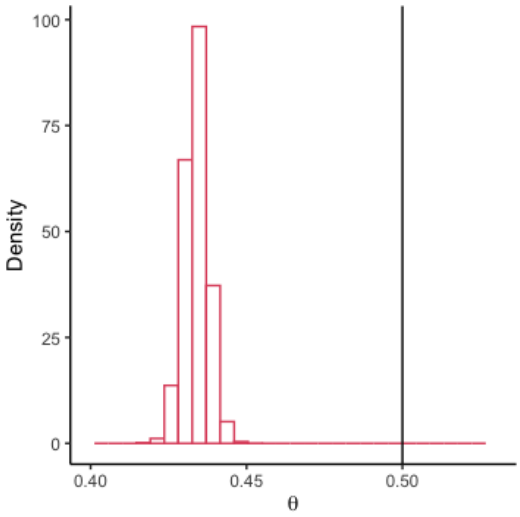
*Can we still infer* $\theta$ *using the model in Equation 2.1 with data coming from Equation 2.2?*

Figure 2.2 shows the corrupted model with sample data coming from the non-linear model:

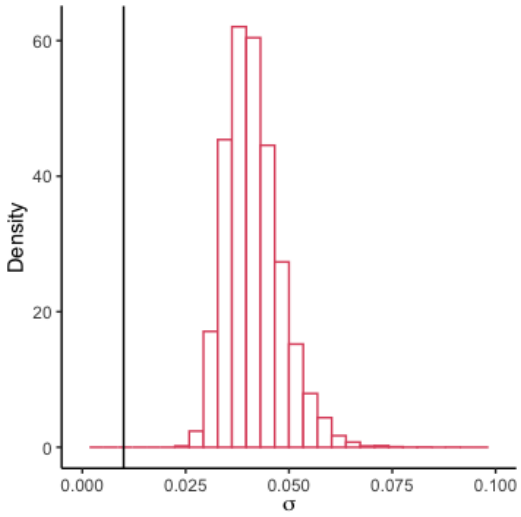
$$y_i = 0.5 \times \frac{x}{1 + \frac{x}{20}} + \varepsilon_i, \qquad \varepsilon_1, \ldots, \varepsilon_n \overset{i.i.d}{\sim} N(0, 0.01^2).$$

We estimate the posterior distribution for $\theta$ and $\sigma$ in model Equation 2.1 with the prior distributions $\theta \sim \mathcal{N}(0, 10)$ and $\log(\sigma) \sim \mathcal{N}(-3, 1)$ and using NUTS[1] in R. It can be seen that the posterior distribution is wrong for both the parameter of interest $\theta$ (underestimated) in Figure 2.3a and for the noise (overestimated around the value 0.05 instead of 0.01) in Figure 2.3b. Even a slight deviation from the linear trend depicted in Figure 2.2 can significantly influence the standard Bayesian estimate.

[1]NUTS stands for the No-U-Turn Sampler variant of Hamiltonian Monte Carlo ([Hoffman and Gelman, 2011, Betancourt, 2017]) and is used for sampling from the posterior distribution in Bayesian statistics.




(a) Bayesian posterior distribution for $\theta$ in red, with the true value shown as the black vertical line.




(b) Bayesian posterior distribution for $\sigma$ in red, with the true value shown as the black vertical line.

Figure 2.3: Example 1 - Corrupted Bayesian Linear Regression.

In the work on robust estimation theory by Huber [2011], a widely reused example: the $\epsilon$-contamination problem (Jewson et al. [2018]) is exploring the impact of contaminated data. In practice, standard statistical approaches such as Maximum Likelihood Estimation (MLE) often perform poorly in the presence of outliers. This phenomenon occurs because the most informative observations often deviate from the model fitted to the majority of the data.

**Example 2** ($\epsilon$-contamination)**.**

*For $\epsilon \in (0, 0.5)$, the $\epsilon$-contamination model with normal distributions is*

$$g(x) = (1-\epsilon)\mathcal{N}(x; \mu_u, \sigma_u^2) + \epsilon\mathcal{N}(x; \mu_c, \sigma_c^2) \tag{2.3}$$

*where $(\epsilon, \mu_u, \sigma_u^2, \mu_c, \sigma_c^2)$ are fixed features of the dGp.*

*This mixture model $g(x)$ consists of two components. The first term $(1-\epsilon)\mathcal{N}(x; \mu_u, \sigma_u^2)$ represents the majority of the data, assumed to follow a normal distribution with mean $\mu_u$ and variance $\sigma_u^2$. The second term $\epsilon\mathcal{N}(x; \mu_c, \sigma_c^2)$*

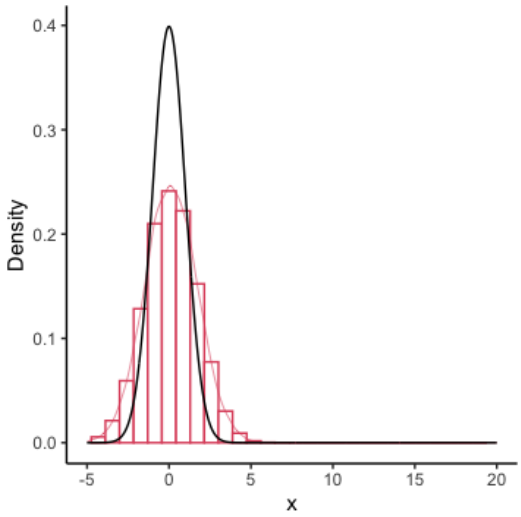



Figure 2.4: Posterior predictive distributions (smoothed from a sample) in red obtained by fitting the baseline model to the $\epsilon$ contamination dataset with $n = 1000$ and the dGp parameters: $(\epsilon = 0.01, \mu_u = 0, \sigma_u^2 = 1, \mu_c = 5, \sigma_c^2 = 5)$. The Gaussian density of the majority of the data, i.e. $\mathcal{N}(0, 1)$ is shown in solid black line.

*represents a smaller sub-population with mean $\mu_c$ and variance $\sigma_c^2$. The parameter $\epsilon$ controls the proportion of this outlying sub-population.*

*Now assume a baseline model given by a standard normal distribution*

$$\mathcal{N}(x; \mu, \sigma^2)$$

*with mean $\mu$ and variance $\sigma^2$. Given $n$ observations from Equation 2.3, we aim to estimate $\theta = (\mu, \sigma^2)$ of the baseline model robustly, even when there is contamination from the outlying sub-population defined by the mixture component.*

In Figure 2.4, the posterior predictive distributions are obtained by fitting a normal model $\mathcal{N}(\mu, \sigma^2)$ to a simulated dataset with $n = 1000$ data points from a normal distribution with $\epsilon$-contamination:

$$0.99 \times \mathcal{N}(0, 1) + 0.01 \times \mathcal{N}(5, 5^2),$$

where the prior distributions are $\mu = \mathcal{N}(0, 1)$ and $\sigma \sim \mathcal{G}(0.001, 0.001)$.

When minimising the KLD, we obtain a density that is centered at $x = (99 \times 0 + 1 \times 5)/100 = 0.05$, i.e. corresponding to 99% of the data but with a larger variance (bigger than 1) since the Bayesian posterior also captures the outlying subgroup.

Example 2 is often referenced to demonstrate the robustness of emerging statistical methods, showcasing a posterior estimate that remains resilient even in the presence of contamination (Miller and Dunson [2019]). In this thesis, the larger uncertainty for the posterior distribution of $\theta$ is not a big concern since the parameter is still adequately inferred. However, this phenomenon can aid us in detecting misspecification in the chosen model.

We choose to delve into more complex forms of model misspecification. Our focus lies on dynamical models constructed based on biophysical mechanistic principles, which are governed by systems of ordinary differential equations (ODEs). Insufficient understanding and incomplete knowledge of a complex dynamical system may result in employing a simplistic model to analyse data derived from the true dGp. For instance, in modelling the electrophysiology of cardiac cells, researchers fit complex mechanistic ion channel models to experimental data. Recent studies have introduced methodologies to incorporate model discrepancy, enhancing inference and predictions ([Rudi et al., 2020, Lei et al., 2020]). Example 3 showcases model discrepancy with three nested ion channel models, reflecting real-world contexts in systems biology.

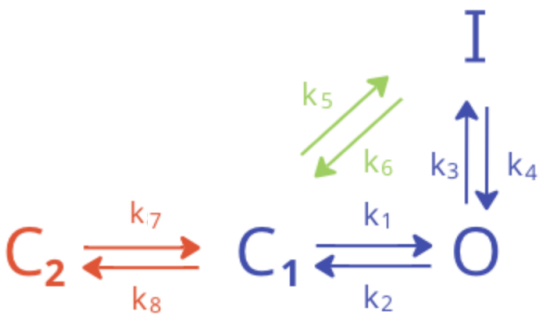



Figure 2.5: Markov model representation of the blue, green, and red models used in the ion channel modelling. The green model includes the blue model and the red model includes the blue and the green models. An example of model discrepancy arises when fitting the blue model to synthetic data generated by either the green or the red model.

**Example 3** (Ion Channel Modelling)**.**

*The following Hodgkin-Huxley model can be employed to fit experimentally recorded currents.*

$$I_{Kr} = G_{Kr} \cdot O(V_m, t) \cdot (V_M - E_K) \tag{2.4}$$

*with the conductance parameter* $G_{Kr}$ *and the open probability* $O(V_m, t)$ *given by the system of equations*

$$\frac{d[I]}{dt} = k_3[O] - k_4[I] + k_5[C] - k_6[I] \tag{2.5}$$
$$\frac{d[O]}{dt} = k_1[C] + k_4[I] - (k_2 + k_3)[O] \tag{2.6}$$
$$\frac{d[C_1]}{dt} = k_2|O| + k_6[I] - [C_1](k_1 + k_5 + k_8) + k_7[C_2] \tag{2.7}$$
$$\frac{d[C_2]}{dt} = k_8[C_1] - k_7[C_2].I \tag{2.8}$$

*The fact state occupancies are probabilities that sum to one tells us that*

$$I = 1 - (O + C1 + C2) \tag{2.9}$$

*which reduces the equation above to only three ODEs instead of four.*

*The rates are voltage-dependent functions, each parameterized by two scalar values of the form* $k_i = A \times \exp(B \times V)$ *except for the rate* $k_6$ *that is obtained using the principle of microscopic reversibility:*

$$k_6 = (k_5 \times k_2 \times k_4)/(k_1 \times k_3), \tag{2.10}$$

*which ensures that each reaction is also in equilibrium when the states are in equilibrium.*

*Below, we depict three embedded models: red, green, and blue, as illustrated in the Markov model (Figure 2.5).*

$$\begin{bmatrix} \frac{dI}{dt} \\ \frac{dO}{dt} \\ \frac{dC_1}{dt} \\ \frac{dC_2}{dt} \end{bmatrix} = \begin{bmatrix} k_3O - k_4I + k_5C_1 - k_6I \\ k_1C_1 + k_4I - (k_2 + k_3)O \\ k_2O - k_1C_1 - k_5C_1 + k_6I + k_7C_2 - k_8C_1 \\ k_8C_1 - k_7C_2 \end{bmatrix} \tag{2.11}$$

*The red model matches precisely the 15-parameter model mentioned earlier in this section. The blue and green models are subsets of the red model, with 11 parameters for the green model and 9 for the blue model. The models are nested, with the blue model contained within the green model within the red model. The blue model is the simplest one and does not account for some specific dynamics of the unknown dGp.*

When the main structure of the model is wrong, a statistician would be left with the obvious answer to change the model. A major difference in mechanistic models is the need for interpretability of the meaningful parameter $\theta$ even if some dynamics of the model have been missed. Mechanistic models are often encoded with a system of differential equations stemming from underlying physical or biological complex processes. Once a model $f(x, \theta)$

is correctly specified, the standard inference techniques can estimate the parameter $\theta$. However, mechanistic models are rarely fully accurate, and mechanistic model misspecification is highly challenging as we do not expect anymore the baseline model to be corrupted but rather the overall structure to be wrong, i.e. missing a differential equation or missing a constitutive meaningful parameter. While statistical uncertainty is an inherent part of parameter estimation in any modelling framework, misspecification in mechanistic models goes beyond statistical noise and involves conceptual errors in the model's representation of the underlying system.

We purposely categorise different misspecification types in this thesis, outlining three forms to be illustrated through examples:

1. (Type I) involves using a slightly modified model based on the dGp, with the right number of parameters. The $\epsilon$-contamination and the non-linear model examples fall under this category, and another example is discussed in Chapter 3 (cf. Example 3.4.2).

2. (Type II) describes a scenario where crucial parameters are omitted from the true model, resulting in an underrepresentation of the actual dynamics (Example 3.4.1 and 3.4.3). In the ion channel modelling example above, this type of misspecification implies that the observations are generated using the green model, while the blue model is utilised for fitting.

3. (Type III) introduces a distinct dimension, where an entirely new set of dynamics is absent from consideration (and parameters as well). Within the context of ordinary differential equations (ODE), this implies the failure to incorporate a new differential equation. This form of misspecification is the most challenging. In Example 3, this scenario would occur when the observations are generated using the red model, while the blue model is employed for fitting.

## 2.3 Bayes under Misspecification

### 2.3.1 Inferential Decisions

Bayesian methods depend strongly on both the prior and the statistical model chosen (Equation 1.1). We let $g(.)$ denote the probability density function of the true distribution of the dGp and assume that the data are random variables $(x_1, ..., x_n)$ on a sample space $\mathcal{X}$. The likelihood model $p(x \mid \theta)$ where $\theta \in \Theta$ is a $d$-dimensional parameter vector with values in $\Theta \subseteq \mathbb{R}^d$ is usually chosen carefully using our knowledge about the dGp.
In scenarios where the decision makers know the family of models from which the data come, referred to as the M-closed view, i.e. where $g = P_{\theta^*}$ for some $\theta^*$, the Bayesian approach to learning is fully justified (Smith and Bernardo [2008]) with the so-called logarithmic loss function or self-information loss (equivalent to the negative log-likelihood).

**Definition 2.1** (The logarithmic loss function)**.** *For the probability density $p(\cdot \mid \theta)$, the log-score (or negative log-likelihood) for a set of i.i.d observations $x_1, ..., x_n$ is:*

$$\ell(\theta, x) = \sum_{i=1}^{n} -\log p(x_i \mid \theta). \tag{2.12}$$

Therefore, we can rewrite the Bayesian posterior in Equation 1.1 with:

$$p(x \mid \theta) = \frac{p(\theta) \exp\{-\ell(\theta, x)\}}{\int_\theta p(\theta) \exp\{-\ell(\theta, x)\}} d\theta.$$

This thesis focuses on the *M-open* world where the model is misspecified.

**Definition 2.2** (*M-closed* and *M-open* world (Smith and Bernardo [2008]))**.** *The M-closed world assumption assumes there exists $\theta_0$ such that the observed i.i.d. data $x_1, \ldots, x_n$ were generated from the model with parameter $\theta_0$, i.e*

$$x_{1:n} \sim p(\cdot \mid \theta_0)$$

*The M-open world assumes that*

$$x_{1:n} \sim g(\cdot)$$

*where it is possible that* $g \neq p(\cdot \mid \theta), \forall \theta \in \Theta$.

### 2.3.2 Bayesian inconsistency under misspecification

We consider a parameter of interest denoted by $\theta$, and let $\Pi_n := p(\theta \mid x_{1:n})$ be the Bayesian posterior of $\theta$ based the datapoints $x_{1:n}$. Under Bayesian updating, a posterior is consistent ([Diaconis and Freedman, 1986, Ghosal et al., 2000]) if the posterior probability concentrates on the true model. For any true parameter value $\theta_0$, as the sample size $n$ tends to infinity, the posterior distribution $\Pi_n$ converges in probability to a Dirac delta function centered at $\theta_0$, i.e. $d(\Pi_n, \delta_{\theta_0}) \to 0$.

Bayesian consistency asserts that, with high probability, the Bayesian posterior concentrates near the true parameter $\theta_0$ value as more data become available. For example, the consistency of the Maximum Likelihood Estimator (MLE) is a fundamental property satisfied under mild regularity assumptions (van der Vaart [1998]) including the identifiability of the parameter $\theta$ and the continuity and differentiability of the likelihood function. For the Maximum Likelihood Estimator (MLE) or other Bayesian estimators denoted as $\hat{\theta_n}$ based on observations, under suitable regularity conditions, as the sample size $n$ grows, the distribution of $\sqrt{n}(\hat{\theta}_n - \theta_0)$ approaches a normal distribution with mean 0 and variance $\sigma^2$.

Therefore, if the dGp is a well-specified model, with enough data, the Bayesian estimator becomes increasingly accurate and converges to the true value $\theta_0$ almost surely:

$$\hat{\theta}_n = \arg\max_{\theta} \sum_{i=1}^{n} \log p_\theta(x_i) \xrightarrow{\mathbb{P}} \theta_0 \tag{2.13}$$

as $n \to \infty$.

Bayesian inconsistency can arise when the model is misspecified, i.e. the assumed statistical model does not accurately represent the underlying data-generating process. Instead of converging to a true value (which does not exist), we converge to the *pseudo-true value* denoted by $\theta^*$:

$$\theta^* = \arg\min_{\theta \in \Theta} \mathrm{KLD}(g \| p(x \mid \theta)) \tag{2.14}$$

which is the parameter that minimised the KLD from the dGp to the wrong model.

This phenomenon becomes notably pronounced when the dGp has heavier tails than the assumed model. In the presence of outliers in the dataset, the limitations of the posterior distribution become apparent as it demonstrates sensitivity with only 1% of contaminated data (cf. Figure 2.4). Notice that $\lim_{p \to 0}(-\log p) = \infty$ so assigning a low probability to any observation will incur a large penalty when using the log-likelihood as the loss function.

However, it is essential to note that this phenomenon is not solely confined to outlier-related situations, as we will explore in this thesis. Standard Bayes methods are often incapable of discerning small deviations from the model (Jewson et al. [2018]). As the data sets grow, even small deficiencies in the model can be disruptive. Sometimes the obtained posterior is sensitive even to seemingly minor deviations between the model and the dGp (Bissiri et al. [2016]).

### 2.3.3 Classical solutions for tackling Model Misspecification in Bayesian statistics

In cases of model misspecification, the decision-maker may contemplate either abandoning the current model in favor of a more comprehensive

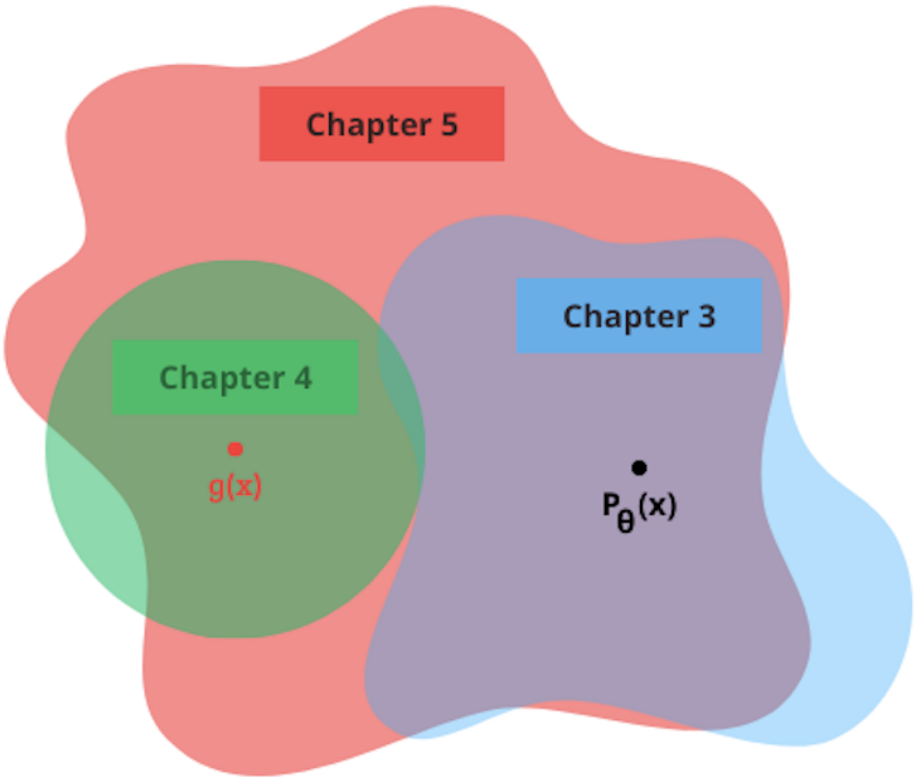



Figure 2.6: The data generating process $g(x)$ (red dot) lies outside the statistical model $p(x \mid \theta : \theta \in \Theta)$ in black, i.e. the model is misspecified. Chapter 3 focuses on classical variational posterior methods to infer $\theta$. In Chapter 4, robust loss functions are employed to generate generalised variational posteriors, aiming to bring the inference process closer to the elusive true dGp, symbolised by the intersection between the green circle and the blue area. Chapter 5 increases the class of model considered by adding a discrepancy function to reduce the disparity between our models and the unknown process, hoping to reach the dGp within the red area.

class of models or transitioning towards data-driven techniques. This thesis specifically addresses the challenge of inferring meaningful parameters within mechanistic models, even when faced with model misspecification. We will focus on methods that mitigate the impact of misspecification by updates in the inference procedure within a Bayesian context. In Chapters 3 and 4, we will keep the model whereas in Chapter 5, we will augment the model class with a discrepancy term as shown in details in Figure 2.6.

In the following sections, we outline classical approaches for addressing model misspecification in Bayesian statistics. This presentation serves as a cohesive literature review, consistent with the mitigation strategies presented in the subsequent sections of the thesis. We emphasise that numerous strategies to address model misspecification fall within the following concepts:

1. Firstly, one approach consists of introducing a tolerance for misspecification, acknowledging the inherent uncertainties in modelling, and allowing for a certain degree of flexibility in capturing the underlying data dynamics (cf. Section 2.3.3.1).

2. Secondly, the deliberate selection of relevant data emerges as a strategy, allowing only the most pertinent information to be used for the inference objective (cf. Section 2.3.3.2).

3. Lastly, more contemporary Bayesian statistical methods recognise the inherent limitations of log-likelihood or KLD-based losses and introduce new losses for more robust and flexible inference frameworks (cf. Section 2.3.3.3).

#### 2.3.3.1 Introduce a tolerance of misspecification

The standard approach to Bayesian inference assumes that the data distribution belongs to the chosen model class. In the presence of slight discrepancies between the assumed model and the unknown dGp, we can still expect robust estimates when the data from the dGp are selected for inference within a specified tolerance. The methods presented below will choose to rely or not on the likelihood considering that only limited information should be used in the inference procedure in the presence of misspecification.

Approximate Bayesian Computation (ABC) is the term given to a collection of algorithms used for calibrating complex simulators ([Csilléry et al., 2010, Marin et al., 2011], Lintusaari et al. [2017]). Suppose $f(\theta)$ is a simulator that models some physical phenomena for which we have observations $x \in \mathbb{R}^n$, and that it takes an unknown parameter value $\theta \in \mathbb{R}^d$ as input and returns output $y \in \mathbb{R}^n$. The Bayesian approach aims to find the posterior distribution $p(\theta \mid x) \propto p(x \mid \theta)p(\theta)$, where $p(\theta)$ is the prior distribution and $p(x \mid \theta)$ is the likelihood function defined by the simulator. ABC algorithms enable the posterior to be approximated using realisations from the simulator, i.e. they do not require explicit knowledge of $p(x \mid \theta)$. The simplest ABC algorithm is the rejection algorithm:

1. Draw $\theta$ from the prior: $\theta \sim p(\theta)$
2. Simulate a realisation from the (deterministic) simulator: $y \sim p(y \mid \theta) = f(\theta)$
3. Accept $\theta$ if and only if $\rho(x, y) \leq h$

where $\rho(,)$ is a distance measure. The tolerance, $h$, controls the trade-off between accuracy and computability. When $h = \infty$ the algorithm returns the prior distribution. Conversely, when $\epsilon = 0$ the algorithm is exact and gives draws from $p(\theta \mid x)$, but acceptances will be rare. In this ABC framework (Beaumont et al. [2002] Wilkinson [2013] Sisson et al. [2018]), the error tolerated depends on the distance measure and the choice of acceptance threshold. If the acceptance threshold is too low, the model may be overly sensitive to small discrepancies between the simulated and observed data and may be rejected even if it is a good approximation to the true data-generating process. On the other hand, if the acceptance threshold is set too high, the model may be overly permissive and may accept poor fits. It means that ABC assumes an imperfect matching between the dGp

and the observations. As an example, the standard likelihood-free ABC rejection algorithm uses a given tolerance $h$ to draw the posterior given the observation $x$, leading to a sample $(\theta, y)$ from the joint distribution proportional to :

$$I(\|y - x\| < h)p(\theta)p(y \mid \theta)$$

where I is the indicator function with $I(Z) = 1$ if Z is true, and $I(Z) = 0$ otherwise. Thus, we discriminate model outputs that are too far from the model observations. The key idea is to acknowledge the discrepancy between the best possible model prediction and the data. We can extend this method by considering a nonuniform acceptance function. Data that match exactly the data process should probably have a bigger weight in the approximation of the posterior. Likewise, we should know when samples from the model $P_\theta$ are the furthest away from the observations. This is the motivation behind using kernel functions. We replace the indicator function $I$ with the kernel function $K_h(u)$ with $u = \|y-x\|$ where $h \geq 0$ corresponds to the scale parameter of the kernel function, and with a kernel symmetric function $K_\epsilon(u)$ such that $K_h(u) \geq 0$, $\forall u$, $\int K(u)du = 1$, $\int uK(u)du = 0$ and $\int u^2K(u)du < \infty$. This leads to the rejection sampling algorithm which gives samples from the joint distribution :

$$p_{ABC}(\theta, y \mid x) \propto p(\theta)K_h(\|y - x\|)p(y \mid \theta). \tag{2.15}$$

If $K_h(u)$ is the uniform kernel, this approach reduces to the rejection algorithm. Accordingly, the Bayesian ABC posterior is:

$$p_{ABC}(\theta \mid x) \propto \int p_{ABC}(\theta, y \mid x)dy. \tag{2.16}$$

**Example 4.** *Suppose that the observed data, $x$ is a single draw from a univariate density function $p(y \mid \theta)$ with $\theta$ a scalar and its prior given by*

*$p(\theta)$. We consider $K_h(u)$ the uniform kernel on $[-h, h]$ with $\|u\| = |u|$. The ABC posterior becomes according to Equation 2.16:*

$$p_{ABC}(\theta \mid x) \propto p(\theta) \int_{-\infty}^{+\infty} K_h(|y - x|) p(y \mid \theta) dy \tag{2.17}$$

$$= \frac{p(\theta)}{2h} \int_{x-h}^{x+h} p(y \mid \theta) dy \tag{2.18}$$

ABC posteriors use a tolerance error as a deliberate strategy to enhance the robustness when dealing with misspecification (Frazier et al. [2017]). Wilkinson (Wilkinson [2013]) pointed out that ABC methods can be considered exact under the assumption of model error.

For deterministic models, we can define $\epsilon$ as the difference between the data and the model run at its best input making it independent of $\theta$, i.e $x = f(\theta) + \epsilon$. From this standpoint, the smoothing kernel $K_h$ serves as the probability density function for this error, such that $\epsilon \sim K_h$, and $h$ represents a scale parameter to be estimated. This approach introduces a level of flexibility to address potential discrepancies, aligning with the principles of Safe Bayes methods (Grünwald and van Ommen [2014]) and would be explored in depth in Chapter 5.

In this approach, a temperature (or annealing) parameter [Mandt et al., 2014] is introduced to diminish the impact of the (misspecified) likelihood in posterior estimation. The resulting posteriors are referred to as Gibbs posteriors [Bissiri et al., 2016, Miller and Dunson, 2019, de Heide et al., 2019]:

$$p(\theta \mid x) \propto p(x \mid \theta)^{\alpha} p(\theta)$$

where $\alpha$ is called the temperature parameter. When $\alpha$ is smaller than 1, then the prior becomes prominent and the data will be less influential. In contrast, when $\alpha > 1$, the log-likelihood is given more prominence and is

helpful to avoid prior misspecification. For appropriately chosen $\alpha$, the Bayes estimator can concentrate at fast rates even under misspecification ([Medina et al., 2022, Ronchetti, 1997]). An illustration, inspired by the approach articulated by Miller and Dunson [2019], is the coarsened posterior. Instead of conditioning precisely on the data, this method conditions on a neighborhood of the empirical distribution, defined by a random variable $R$ representing the tolerance error term expressed as:

$$p(\theta \mid \rho(y, x) < R)$$

with $\rho$ a statistical distance. With $R \sim \exp(\alpha)$, we obtain the Gibbs posterior (cf. Chapter 4). The ABC and annealed offer the advantage of incorporating the likelihood so that the model is expected to retain some fidelity in capturing aspects of the true distribution.

An alternative but closely related approach including a tolerance rate in the inference procedure is known as History Matching (HM) ([Craig et al., 1997, Williamson et al., 2013, Andrianakis et al., 2015]). The HM approach can identify and rule out regions of the parameter space for which we do not have a good match between the observed data and the simulated data. This approach seeks to find values of the model inputs that could not possibly have produced the data. The parameter space is restrained through an iterative procedure that discards $\theta$ based on an implausibility measure that quantifies the mismatch between the model predictions and the observed data known as the discrepancy. This measure commonly includes a discrepancy term, capturing the residual differences between the model's predictions and the actual observations (Kennedy and O'Hagan [2001], Craig et al. [1997]). If we take again the corrupted linear regression

(Example 1), we consider the following implausibility measure:

$$I(\theta) = \max_i \frac{|y_i - \theta x_i|}{\sqrt{\tau^2 + \sigma^2}} \tag{2.19}$$

with $\sigma = 0.01$ the observation error standard deviation and $\tau$ the model discrepancy standard deviation to be chosen. As seen in Figure 2.7, we discard values for the inferred parameter $\theta$ when the observed data do not match the fitted model via the implausibility measure within a threshold (defined at 3 here). This method requires to define the error term $\tau$. Opting for strictness (setting a low value for $\tau$) could lead to complete disregard of the actual value, as illustrated in Figure 2.7 (left plot). HM operates iteratively with different thresholds leading to non-implausible regions (below the threshold), i.e. regions of the space not ruled out for $\theta$. The philosophy of history matching is not to find the best input value for the model but to explore the space of non-implausible values for the model parameters. This strategy may detect if the model is misspecified when the error term is carefully chosen.

In conclusion, the proposed methods strive for robust estimates when faced with slight discrepancies within a predefined tolerance. We will explore their effectiveness in Chapter 4, particularly under higher degrees of misspecification than traditionally examined.

#### 2.3.3.2 Selective Decision-Making Amid Limited Information

If the data generated by a model deviates from the observations of the unknown data-generating process (dGp), it may indicate inaccuracies in the likelihood or potential errors in the observed data itself. In practice, the data can be affected by corruption, such as the presence of outliers, and may also be partially uninformative, leading to a suboptimal posterior estimate. Several authors ([Sisson et al., 2018, Priddle et al., 2019]) have proposed to

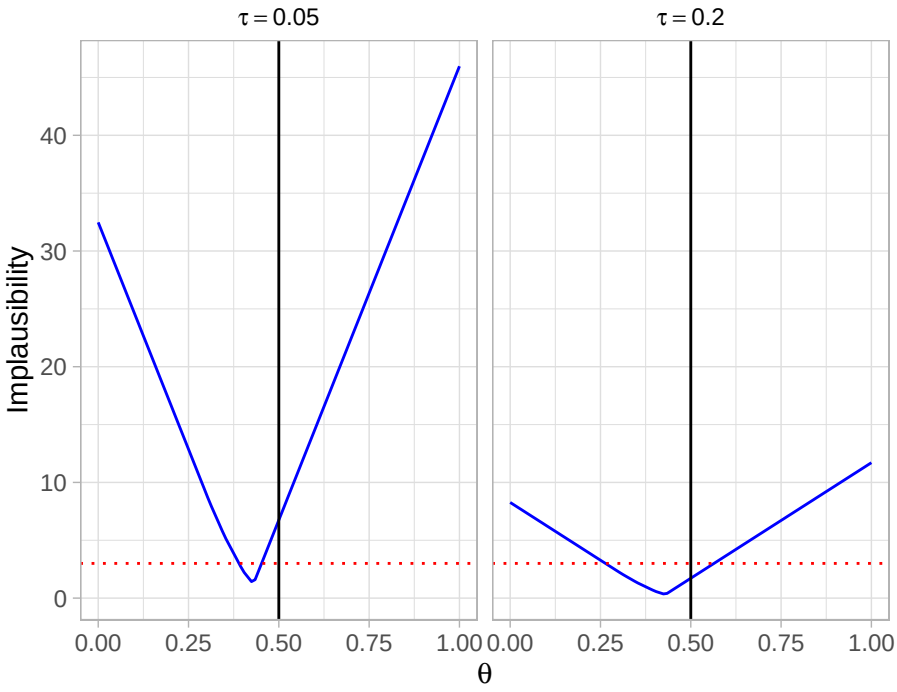



Figure 2.7: History Matching for the Bayesian Linear Regression. Implausible measure for different values of $\tau$. The red dotted line indicates the threshold beyond which values for $\theta$ are discarded. The true value $\theta = 0.5$ is discarded for $\tau = 0.05$ and is accepted for $\tau = 0.2$.

use summary statistics denoted as $s(x)$ moving from matching data $\|y - x\|$ to summary statistics $\|s(y) - s(x)\|$ to improve the quality of the inherent inference despite the loss of information. The use of summary statistics has proven to be a useful tool, especially for ABC posteriors ([Frazier et al., 2017, Frazier, 2020]) where quantities such as moments or sample quantiles are used in the simplest case highlighting the computational benefits in using lower dimensional data.

**Example 5.** *Suppose that the observed data, $x$ are $n$ independent draws from a univariate $\mathcal{N}(\theta, \sigma^2)$ distribution with $\sigma > 0$ known. We use the Gaussian Kernel $K_h(u) = \frac{1}{\sqrt{2\pi h^2}} \exp(-\frac{1}{2h^2}u^2)$, for $h \geq 0$. The most natural summary statistics is the mean of the observed samples, i.e. $s(x) = \sum_{i=1}^{n} x_i$.*

*The ABC likelihood according to Equation 2.16 is :*

$$p_{ABC}(s(x)|\theta) = \int_{-\infty}^{+\infty} K_h(\|s(y) - s(x)\|)p(s(y) \mid \theta)ds(y) \tag{2.20}$$

$$= \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}h} \exp(-\frac{(s(y) - s(x))^2}{2h^2}) \frac{\sqrt{n}}{\sqrt{2\pi}\sigma} \exp(-\frac{n(s(y) - \theta)^2}{2\sigma^2})ds(y) \tag{2.21}$$

$$\propto \exp(-\frac{(\theta - x)^2)}{2(\sigma^2/n + h^2)}) \tag{2.22}$$

*The true likelihood for which the observations $s(x) \sim \mathcal{N}(\theta, \sigma^2/n)$ is replaced with a higher variance under the ABC approximation $s(x) \sim \mathcal{N}(\theta, \sigma^2/n + h^2)$. If $h \to 0$, then both likelihoods are equivalent. As $h$ gets larger, we take into account the potential model error by increasing the uncertainty.*

If we consider the misspecified scenario (Example 2), a summary statistics based on the mean is sensitive to the contamination with a direct association to the size of the subgroup of outliers, i.e. $n - \epsilon \times 100$ of the subgroup of outliers.

In this thesis, the misspecification discussed is not primarily associated with the data acquired from the real world (unknown dGp). Rather, the primary concern revolves around errors intrinsic to the model itself. This explains why examples studied in the literature, particularly those involving contamination, may not be sufficient in scenarios of higher misspecification to ensure robust inference.

#### 2.3.3.3 Beyond the logarithmic loss

As soon as the model assumptions are not honoured, we have serious difficulties to correctly infer the parameter $\theta$ given observations $x_{1:n}$. The link between the observed data and parameters is broken because of the model inaccuracy. Unless we can correct the model, we should not expect asymptotic convergences of the posterior for $\theta$. One proposed solution is to

move away from the standard log-likelihood loss, and instead minimise a new loss function to learn the parameter $\theta$ (Jewson et al. [2018]). New loss functions may be able to provide more robust M-estimators (cf. Section 2.4) and a range of statistical distances have been explored with this goal in mind, such as Stein discrepancy (Matsubara et al. [2021]) or Maximum Mean Discrepancy (MMD) (Chérief-Abdellatif and Alquier [2019]). Briol et al. [2019] proved for example that MMD distance-based estimators are robust to some forms of model misspecification, and some trade-off between statistical efficiency and robustness can be achieved through the choice of the kernel.

Knoblauch et al. [2019] challenge the traditional Bayesian inference paradigm and generalise it in a way that can confront model misspecification. They derive a generalisation of the Bayesian inference problem with a new paradigm called Generalised Variational Inference (GVI) (cf. Chapter 4). Following the terminology established by Knoblauch et al., a Bayesian inference method seeks to solve the following optimization problem:

$$P(\ell_n, D, \mathcal{Q}) : q^*(\theta) = \underset{q(\theta)\in\mathcal{Q}}{\arg\min}\, \mathbb{E}_{q(\theta)}\left[\sum_{i=1}^{n} \ell(\theta, x_i)\right] + D(q(\theta)||p(\theta)), \quad (2.23)$$

with :

1. a **loss** $\ell_n$ defining the target parameter for inference relative to the sample distribution of the observations. We only consider additive losses in the thesis, i.e $\ell_n(\theta, x_{1:n}) = \sum_{i=1}^{n} \ell(\theta, x_i)$.

2. a statistical **divergence** $D : \mathcal{Q} \times \mathcal{Q} \to \mathbb{R}_+$ regularising the posterior with respect to the prior distribution $p(\theta)$ of the parameter $\theta$.

3. a tractable family of distributions for the posterior $\mathcal{Q} \subseteq \mathcal{P}(\Theta)$ with $\mathcal{Q} = \{q(\theta) : \int q(\theta)d\theta = 1\}$ and $\mathcal{P}(\Theta)$ the space of Borel probability

measures on $\Theta$.

The previous equation recovers standard VI posteriors defined in Equation 1.3 using the KLD and the log-likelihood loss (Definition 2.1). Most importantly, the generalised variational posterior is closely related to the standard Bayesian posterior $p(\theta \mid x_{1:n})$ defined previously as:

$$p(\theta \mid x_{1:n}) = \frac{p(x_{1:n} \mid \theta)p(\theta)}{\int_\Theta p(x_{1:n} \mid \theta)p(\theta)d\theta}.$$

**Theorem 2.1.** *If we define the loss* $\ell_n(\theta, x_{1:n}) = \sum_{i=1}^n -\log p(x_i|\theta)$*, the divergence* $D = KLD$*, and the family of distributions* $\mathcal{Q} = \mathcal{P}(\Theta)$*, and if* $Z = \int_\Theta \exp\{-\ell_n(\theta, x_{1:n})\}p(\theta)d\theta < \infty$*, then* $P(l_n, D, \mathcal{Q})$ *is the standard Bayesian posterior.*

*Proof.*

We can rewrite the objective of Equation 2.23 as:

$$\begin{aligned} q^*(\theta) &= \underset{q\in\mathcal{P}(\Theta)}{\arg\min}\left\{\int_\Theta \left[\log(\exp(l_n(\theta, x_{1:n}))) + \log\frac{q(\theta)}{p(\theta)}\right]q(\theta)d\theta\right\} \\ &= \underset{q\in\mathcal{P}(\Theta)}{\arg\min}\left\{\int_\Theta \log\left(\frac{q(\theta)}{p(\theta)\exp(-l_n(\theta, x_{1:n}))}\right)q(\theta)d\theta\right\}. \end{aligned}$$

As we are only interested in finding the minimiser $q^*(\theta)$ (and not the objective value), it also holds that for any constant $Z > 0$, the above is equal to:

$$\begin{aligned} q^*(\theta) &= \underset{q\in\mathcal{P}(\Theta)}{\arg\min}\left\{\int_\Theta \log\left(\frac{q(\theta)}{p(\theta)\exp(-l_n(\theta, x_{1:n}))Z^{-1}}\right)q(\theta)d\theta - \log Z\right\} \\ &= \underset{q\in\mathcal{P}(\Theta)}{\arg\min}\left(\mathrm{KLD}\left(q(\theta)||p(\theta)\exp(-\ell_n(\theta, x_{1:n}))Z^{-1}\right)\right). \end{aligned}$$

Lastly, setting $Z = \int_\theta \exp\{-l_n(\theta, x_{1:n})\}p(\theta)d\theta$, and noting that the KLD is minimised uniquely if its two arguments are the same, we conclude that $q^*(\theta) = p(\theta \mid x_{1:n})$. □

This framework can provide a more flexible and adaptive approach to Bayesian inference, allowing for a more realistic representation of the underlying data-generating process. This approach has yielded more reliable and robust results ([Knoblauch, 2019, Knoblauch et al., 2018, Husain and Knoblauch, 2022, Altamirano et al., 2023, Dellaporta et al., 2022]) addressing various challenges associated with model misspecification, such as accommodating outliers, handling heavy-tailed or skewed distributions, capturing complex dependencies whilst still incorporating prior information and expert knowledge. Notice that the parameter $\theta$ is only affected by the loss function in the GVI problem. Therefore, alternative loss functions to the log score will change the target parameter for inference to gain robustness against model misspecification (modularity rule in Knoblauch et al. [2019]). Chapter 4 of this thesis is grounded in the GVI framework, where we employ a curated selection of robust loss functions. As an example, the $\alpha$-Divergence (Definition 4.3.1) exhibits better robustness for the $\epsilon$-contamination problem as shown in the next Section in Figure 2.9.

## 2.4 Review on Robust Measures

### 2.4.1 Deterministic models

This section delves into commonly used robust measures in the literature to understand and evaluate the robustness of inference frameworks when dealing with model misspecification. Defining robustness proves challenging and is highly dependent on the specific context. Nevertheless, several common robust measures provide essential concepts for assessing the sensitivity of the inference to model error. These measures will be examined within the context of this thesis, and we will address any limitations and propose potential improved approaches for assessing robustness in the next

chapters.

Specifically addressing outliers, as exemplified in Example 2, robust measures like the median or the Winsorized mean offer alternatives that are less sensitive to extreme values. The Winsorized mean is a robust measure of central tendency that reduces the impact of outliers by replacing the smallest and largest values in a dataset with specified percentiles before calculating the mean. This approach helps mitigate the influence of extreme values on the overall average. These measures assess the ability of an estimator to maintain accuracy and reliability when deviations from the true model happen.

The finite sample breakdown point (Definition 2.3) is a global measure of robustness corresponding to the minimum proportion of observations in the sample that need to be perturbed to make the distance between the estimates based on the original and contaminated samples arbitrarily large. For example, it is the percentage of outliers in a data set causing a robust estimator to break down and produce unreliable results (cf. Example 6). The asymptotic breakdown point is usually given by the limit of the finite sample breakdown point as the data size $n$ goes to infinity.

**Definition 2.3** (Breakdown Point)**.** *Denote the estimator as $T$ and the distribution of the data as $P$. Define $b(\epsilon) = \sup_x |T(P) - T(P_\epsilon)|$, where $P_\epsilon = (1-\epsilon)P + \epsilon$. The breakdown point of an estimator is denoted as $\epsilon^*$ and is defined as $\epsilon^* = \inf\{\epsilon : b(\epsilon) = \infty\}$.*

**Example 6** (Breakdown Point)**.**

1. *The value of the mean $\frac{1}{n}\sum_1^n x_i$ can be changed by an arbitrarily large amount, simply by changing one of the data points. Therefore, the finite breakdown point is just $\frac{1}{n}$ and the asymptotic breakdown point is zero.*

2. *The median can tolerate extreme values either on the left or the right side. The finite sample breakdown point is [(n - 1)/(2n)] and the*

*asymptotic breakdown point is* $\frac{1}{2}$.

Typically, a higher breakdown point implies that the estimation procedure is more robust to outliers or errors because it can tolerate a larger proportion of contaminated data before the accuracy of the estimates is compromised. It is interesting to notice the close connection between the definition of this measure and the contamination described in example 2 where we evaluate the sensitivity of the contamination on the final estimate (or posterior distribution).

M-estimation (Magnusson [1975]) is a broad class of statistical methods used for estimating the parameters of a statistical model. M-estimators seek to find the values of parameters that minimise or maximise an objective function, often derived from a likelihood function or another criterion related to the sample data. We can consider some robust M-estimators to mitigate the impact of outliers in the inference. Prominent examples of M-estimators can be the Huber (Definition 2.4) and Tukey (Definition 2.5) losses described below.

**Definition 2.4** (Huber loss)**.**

$$\ell_\delta(x) = \begin{cases} \frac{1}{2}x^2, & \text{if} \quad |x| < \delta \\ \delta|x| - \frac{1}{2}\delta^2, & \textit{otherwise} \end{cases}$$

**Definition 2.5** (Tukey loss)**.**

$$\ell_\delta(x) = \begin{cases} \frac{\delta^2}{6}(1-(1-\frac{x}{\delta})^2)^3, & \text{if} \quad |x| < \delta \\ \frac{\delta^2}{6}, & \textit{otherwise} \end{cases}$$

The log-likelihood loss discussed before transforms into a least square function when the model is Gaussian. In Figure 2.8, the Huber and Tukey losses can replace the traditional least squares cost function to de-emphasise outliers since the residuals are much smaller. In both estimators, the tunable

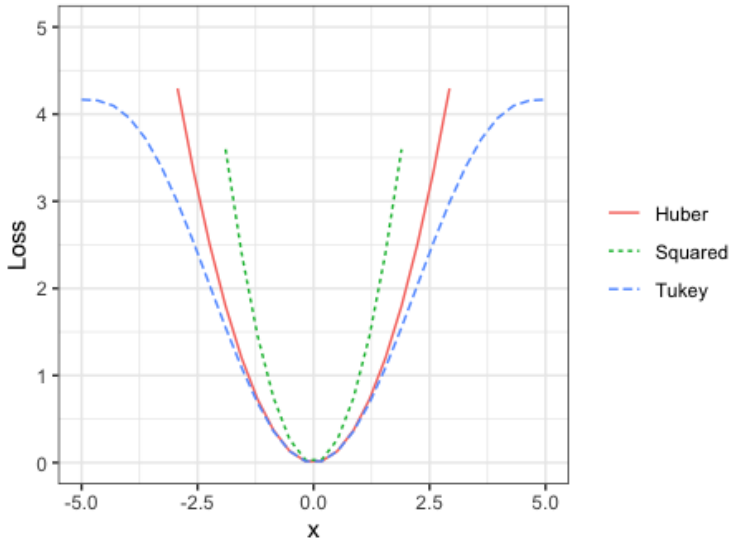



Figure 2.8: Comparison of Huber, least square, and Tukey loss functions.

constant, denoted as $\delta$ allows for the adjustment of the threshold that defines what qualifies as an outlier in the estimation procedure.

Lastly, the influence function is used to quantify the sensitivity of the estimator denoted as $T$ to small changes in the distribution around a single data point $x$ (Definition 2.6). $\epsilon$ represents the infinitesimal perturbation applied to the distribution in the definition.

**Definition 2.6** (Influence function)**.** *The influence function (IF) of a functional $T$ at a distribution $P$ is given by:*

$$IF(x; T, P) = \lim_{\epsilon \to 0} \frac{T((1-\epsilon)P + \epsilon\delta_x) - T(P)}{\epsilon}$$

*$\delta_x$ denotes a discrete distribution that assigns probability 1 to the point $x$.*

The influence function provides insights into the robustness of the estimator at different data points and helps assess its behaviour under small perturbations of the distribution.

### 2.4.2 Statistical models

In the previous section, we presented robustness measures for estimators, primarily within a frequentist framework. They are not the only existing measures but the best-known and widely used measures, especially the influence function and the breakdown point.

In the context of Bayesian statistics, we establish the measure of influence by considering distances between posteriors. One can evaluate the influence of the $k^{th}$ observation on the posterior distribution by removing it from the observation set and estimating the posterior distribution using the remaining observations ([Peng and Dey, 1995, Kurtek and Bharath, 2014]).

Let $x_k$ denote the $k^{th}$ observation, and $x(-k)$ represent the set of observations obtained by excluding the $k^{th}$ one. Given the baseline posterior $p_0 = p(\theta \mid x_{1:n})$, let us remove one data point $x_k$ and obtain the posterior $p_k = p(\theta \mid x(-k))$. The influence function proposed by Kurtek and Bharath [2015] uses the geodesic distance under the Fisher–Rao metric (SI Amari et al. [1987]). The influence with the respective likelihood $f_0(x|\theta)$ and $f_k(x(-k)|\theta)$ with $x$ corresponding to the entire dataset $x_{1:n}$ is defined by:

$$IF(k) = d_{FR}(p_0, p_k) = \left[ \int_\Theta \frac{1}{f_0(x_k|x(-k),\theta)} p_0 d\theta \right]^{-1/2} \int_\Theta \left[ \frac{f_k(x|\theta)}{f_0(x|\theta)} \right]^{1/2} p_0 d\theta. \tag{2.24}$$

Given a sample from the baseline posterior density, $\theta_1, ..., \theta_N$, Kurtek and Bharath [2015] propose the following Monte Carlo estimate of $IF(k)$:

$$\hat{I}(k) = \cos^{-1} \left[ \frac{b}{N} \sum_{i=1}^{N} a_i \right] \tag{2.25}$$

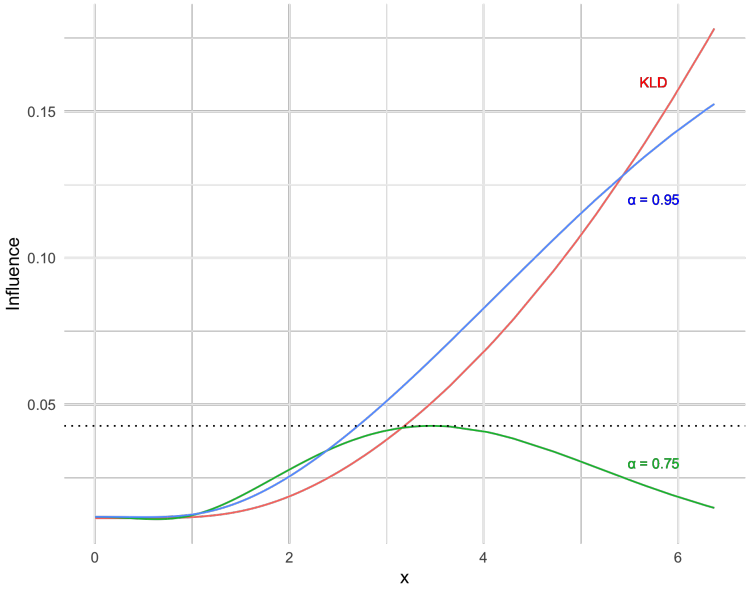



Figure 2.9: The influence, as described in Kurtek and Bharath [2015], of removing one out of 1000 observations from the $\epsilon$-contaminated distribution when fitting $\mathcal{N}(\mu, \sigma^2)$ under the KLD divergence and the Alpha-Divergence with two values for $\alpha$ is depicted. For $\alpha = 0.75$, the influence is bounded by the black horizontal line.

with

$$a_i = \left[\frac{f_k(x|\theta_i)}{f_0(x|\theta_i)}\right]^{1/2} \quad b = \left[\frac{1}{N}\sum_{i=1}^{N}\frac{1}{f_0(x_k|x(-k),\theta_i)}\right]^{-1/2}$$

Generating only one posterior sample is sufficient to assess the influence measure for all observations, making this approach computationally tractable. Figure 2.9 demonstrates a rise in influence for observations in the tails with the Kullback Leibler divergence, contrasting with a decrease in influence for outlying observations under the $\alpha$-robust divergence for the $\epsilon$-contamination problem. In this setup, we compute the Fisher-Rao distance between two Gaussian posteriors and the Monte Carlo estimate is therefore straightforward. The $\alpha$ divergence shows concave influence functions, particularly with lower values of $\alpha$.

The robustness of various methods has been extensively studied when considering contaminated data, as described above. In this thesis, we will meticulously select mechanistic models and subject them to various per-

turbations. Our objective is to assess whether small structural misspecifications in the model have significant consequences for the robustness of different inference methods. The thesis will ideally help us understand the sensitivity of outcomes to changes in model assumptions, providing valuable insights into solutions to make safer inferences.

## 2.5 Considerations

If the model is a good approximation of the dGp, the standard Bayesian inference strategy is often successful, with an optimal posterior when the true unknown distribution belongs to the class of the fitted model. However, standard Bayesian methods are often incapable of discerning small deviations from the model. As the data set grows, even small deficiencies in the model can be disruptive, and therefore Bayesian posteriors may become highly sensitive to seemingly minor deviations from assumptions made about the model. Our objective is to ensure the reliability and accuracy of inferential methods even in the presence of misspecification. To achieve this goal, we elaborate on various approaches documented in the literature to tackle issues associated with model misspecification.

Both Approximate Bayesian Computation and History Matching can provide useful information in the presence of misspecified models, particularly by evaluating which parameters should be excluded in the inference process. They can rely on summary statistics based on the researcher's knowledge or threshold of some user-specified loss. However, these methods may lead to a loss of information, resulting in an incomplete representation of the data and discarding valuable insights. Other methods discussed in this chapter opt to make slight modifications to the log score function, also known as the annealed posterior, to mitigate the impact of the likelihood on the final posterior estimate. Alternatively, some approaches entirely shift to new

loss functions for more robust inference. The GVI framework establishes a framework for incorporating any new loss functions into the optimization variational procedure. These new losses may demonstrate robustness properties, as elaborated in Chapter 4.

When examining classical robust measures, which assess the effects of deviations from underlying statistical assumptions, we discover that model misspecification is primarily explored through the perspective of contaminating a distribution or when the model contains minor errors. They often assume correct model specification (Type 1), but what about their effectiveness when the model is structurally wrong (Type 2 and Type 3)? If a linear regression model is used to analyse data with a nonlinear relationship (Example 1), robust measures studied above will not detect the structural misspecification. In cases where a dynamic is missed, examining the influence of a single data point or a small subset may not be particularly informative.

# Chapter 3

# Auto-Encoding Variational Bayes for physics-informed models

**Summary:** This chapter describes an automatic variational inference method for inference in dynamical systems. We use the particular variational auto-encoders architecture to perform inference for ordinary differential equation models. The methodology leverages differentiable ODE solvers, showcasing the potential to enhance inference within physics-informed models. We delve into its application and implications in situations where model specifications may deviate from the actual underlying processes.

This chapter is organised as follows. In Section 3.1, we offer a technical introduction to variational inference, and in Section 3.3, we provide an overview of automatic differentiation. In Section 3.2, we present the variational autoencoder framework within the context of generative models based on ordinary differential equation models. We utilise PyTorch automatic differentiation to facilitate the backpropagation of mechanistic models without the conventional reliance on neural networks, presenting our results for well-specified and misspecified models.

## 3.1 Variational Inference

### 3.1.1 Variational Bayes

Let $x$ denote the data and $p(x \mid \theta)$ the likelihood function for a postulated model, with $\theta \in \Theta$ the vector of model parameters to be estimated. Let $p(\theta)$ be the prior. Bayesian inference encodes all the available information about the model parameter $\theta$ in its posterior distribution with density given by Equation 1.1. Bayesian inference typically requires computing expectations with respect to the posterior distribution. Determining these expectations can pose challenges, due to the intractability of the posterior density $p(\theta \mid x)$ as the normalising constant $p(x)$ is often unknown. In these situations, Bayesian inference is often performed using sampling methods such as Markov Chain Monte Carlo (MCMC) which generates samples from the posterior $p(\theta \mid x)$. Variational Bayes (VB) solves the Bayesian inference problem by solving an optimization problem in contrast with Markov Chain Monte Carlo (MCMC) where we approximate the posterior by sampling from it. VB seeks to approximate the posterior distribution by selecting an optimal distribution $q(\theta)$ from some tractable family of distributions $\mathcal{Q}$, such as the family of Gaussian distributions (cf. Section 3.1.2). The best VB approximation $q^*(\theta) \in \mathcal{Q}$ is found by minimising the Kullback-Leibler divergence (KLD) from $q(\theta)$ to $p(\theta \mid x)$ ([Ormerod and Wand, 2010, Blei et al., 2016]) :

$$q^*(\theta) = \arg\min_{q \in \mathcal{Q}} \left\{ \mathrm{KLD}(q(\theta) \| p(\theta \mid x) := \int q(\theta) \log \frac{q(\theta)}{p(\theta \mid x)} d\theta \right\}. \quad (3.1)$$

Any posterior expectations can then be performed by replacing the intractable posterior $p(\theta \mid x)$ with the tractable VB approximation $q^*(\theta)$. We note that minimising the Kullback-Leibler Divergence (KLD) in Equa-

tion 3.1 is equivalent to maximising the lower-bound defined below.

$$\begin{aligned}
\mathrm{KLD}(q(\theta)||p(\theta \mid x)) &= \int_{\theta} q(\theta) \log \frac{q(\theta)}{p(\theta \mid x)} d\theta && (3.2)\\
&= \int_{\theta} q(\theta) \log \frac{q(\theta)p(x)}{p(x,\theta)} d\theta \quad \textit{(Conditional probability (CP))} && (3.3)\\
&= \log p(x) + \int_{\theta} q(\theta) \log \frac{q(\theta)}{p(x,\theta)} d\theta \quad \textit{(integretation over } \theta) && (3.4)\\
&= \log p(x) - \mathcal{LB}(q). \quad \textit{(definition of the ELBO)} && (3.5)
\end{aligned}$$

where

$$\mathcal{LB}(q) = \int_{\Theta} q(\theta) \log \frac{p(\theta, x)}{q(\theta)} d\theta$$

is called the Evidence Lower Bound (ELBO).

From the above, we can see that finding the distribution $q(\theta)$ that maximises the ELBO is equivalent to finding the distribution $q(\theta)$ that minimises the KLD to the posterior. The difference between the ELBO and the KLD is precisely the evidence $\log p(x)$ (marginal likelihood) in Equation 3.5, which is the quantity that the ELBO bounds.

$$\log p(x) \geq \mathcal{LB}(q) = \mathbb{E}_{q(\theta)}[\log p(x,\theta)] - \mathbb{E}_{q(\theta)}[\log q(\theta)]. \tag{3.6}$$

Intuitively, the first term $\mathbb{E}_{q(\theta)}[\log p(x,\theta)]$ in Equation 3.6 encourages variational distributions to assign high mass on configurations of the latent variables that effectively explain the given observations. The second term $-\mathbb{E}_{q(\theta)}[\log q(\theta)] := \mathcal{H}(q(\theta))$ is the entropy of the variational distribution that encourages variational distributions to be diverse and spread their mass across multiple configurations. This term regularises the problem and avoids data overfitting.

### 3.1.2 Mean Field Variational Inference

If we allow $\mathcal{Q}$, the set of approximations considered in the optimization to be unconstrained, the solution to Equation 3.1 is $q^*(\theta) = p(\theta \mid x)$. This solution is useless as it is itself intractable. Depending on the constraint imposed on the class $\mathcal{Q}$, VB algorithms can be categorised into two classes: Mean Field VB (MFVB) and Fixed Form VB (FFVB).

Mean Field Variational Inference (MFVI) has its origins in the mean-field theory of physics (Opper and Saad [2001]) and considers an approximating family $\mathcal{Q}$ that includes all factorizable densities :

$$\mathcal{Q} = \Big\{ q(\theta) : q(\theta) = \prod_{i=1}^{d} q_i(\theta_i) \Big\}.$$

When we assert that the variational family has this factorized form, we can derive a coordinate ascent algorithm and obtain a fully factorized variational distribution that maximises the ELBO via simple iterative updates. For more details on the mean-field approximation and the so-called naive mean-field algorithm and its geometrical interpretation, we refer the reader to Bishop [2006] and Wainwright and Jordan [2007].

FFVB assumes a fixed parametric form for the variational approximation density $q$, i.e. $q = q_\phi$ belongs to some class of distributions $\mathcal{Q}$ indexed by a vector $\phi$ called the variational parameter. For example, we choose $q_\phi$ to be a Gaussian distribution with mean $\mu$ and covariance matrix $\Sigma$. The variational parameter set is $\phi = (\mu, \Sigma)$. FFVB finds the best $q_\phi$ in the class $\mathcal{Q}$ by optimizing the lower bound :

$$\mathcal{LB}(\phi) = \mathbb{E}_{q_\phi}\left[ \log\Big( \frac{p(\theta)p(x \mid \theta)}{q_\phi(\theta)} \Big) \right] = \mathbb{E}_{q_\phi}[h_\phi(\theta)], \tag{3.7}$$

with

$$h_\phi(\theta) := \log\Big(\frac{p(\theta)p(x \mid \theta)}{q_\phi(\theta)}\Big).$$

The ELBO $\mathcal{LB}(\phi)$ is now a function of the variational parameters $\phi$, which we can optimize using gradient-based optimization methods.

### 3.1.3 Stochastic Variational Inference

The coordinate ascent algorithm used in MFVI is inefficient for large data sets because it requires evaluating the likelihood for the entire dataset for each update. An alternative approach to coordinate ascent is gradient-based optimization, where the ELBO gradients are updated at each iteration. This perspective motivates the foundation for the scalability of variational inference through the utilisation of Stochastic Variational Inference (SVI) ([Hoffman et al., 2013, Hoffman and Blei, 2014]) and Black Box Variational Inference (BBVI) (Ranganath et al. [2013]). In SVI, we build a noisy estimate of the gradient $\nabla_\phi \mathcal{LB}(\phi)$ by sampling (Angelino et al. [2016]) a batch of the data and then use stochastic gradient descent ([Zhang et al., 2019, Robbins and Monro, 1951]) so that in every iteration, one randomly selects mini-batches to obtain a stochastic estimate of the ELBO.

A convenient form of the ELBO gradient can be obtained as follows:

$$\nabla_\phi \mathcal{LB}(\phi) = \nabla_\phi \mathbb{E}_{q_\phi(\theta)}\Big[\log\Big[\frac{p(x,\theta)}{q_\phi(\theta)}\Big]\Big], \tag{3.8}$$

$$= \int_\Theta \nabla_\phi\Big(\log\Big[\frac{p(x,\theta)}{q_\phi(\theta)}\Big] q_\phi(\theta)\Big) d\theta, \quad \textit{(Leibniz's rule)} \tag{3.9}$$

$$= \int_\Theta \log\Big[\frac{p(x,\theta)}{q_\phi(\theta)}\Big] \nabla_\phi(q_\phi(\theta)) + q_\phi(\theta)\nabla_\phi(\log\Big[\frac{p(x,\theta)}{q_\phi(\theta)}\Big]) d\theta. \tag{3.10}$$

The second term can be simplified since $p(x, \theta)$ is independent of $\phi$.

$$\mathbb{E}_{q_\phi(\theta)}\Big[\nabla_\phi \log\Big[\frac{p(x,\theta)}{q_\phi(\theta)}\Big]\Big] = -\mathbb{E}_{q_\phi(\theta)}\Big[\nabla_\phi \log q_\phi(\theta)\Big]$$

Note the log-derivative trick (LDT):

$$q_\phi(\theta)\nabla_\phi(\log q_\phi(\theta)) = \nabla_\phi(q_\phi(\theta)),$$

so that the ELBO gradient becomes:

$$\nabla_\phi \mathcal{LB}(\phi) = \mathbb{E}_{q_\phi(\theta)}\Big[h_\phi(\theta) \times \nabla_\phi \log q_\phi(\theta)\Big] - \mathbb{E}_{q_\phi(\theta)}\Big[\nabla_\phi \log q_\phi(\theta)\Big]\Big] \quad (3.11)$$

$$= \mathbb{E}_{q_\phi(\theta)}\Big[h_\phi(\theta) \times \nabla_\phi \log q_\phi(\theta)\Big] - \nabla_\phi \int q_\phi(\theta) d\theta. \quad (3.12)$$

$$= \mathbb{E}_{q_\phi(\theta)}\Big[h_\phi(\theta) \times \nabla_\phi \log q_\phi(\theta)\Big]. \quad (3.13)$$

The second term in Equation 3.12 is obtained with the log-derivative trick again and is equal to zero since the density function integrates to 1. The gradient $\nabla_\phi \mathcal{LB}(\phi))$ has then a particular form often referred to as the score-function gradient. It follows from the above that, if we generate $\theta_i \sim q_\phi(\theta)$, $\widehat{\nabla_\phi \mathcal{LB}(\phi))} = h_\phi(\theta_i) \times \nabla_\phi \log q_\phi(\theta_i)$ is an unbiased estimator of the gradient $\nabla_\phi \mathcal{LB}(\phi)$. If we use $\widehat{\nabla_\phi \mathcal{LB}(\phi))}$ instead of $\nabla_\phi \mathcal{LB}(\phi)$, we have a stochastic optimization algorithm (Kingma and Ba [2014]) following the basic steps given in the Algorithm below.

**Algorithm 1** (Basic stochastic gradient descent FFVB algorithm)**.**

- *Initialize* $\phi^{(0)}$ *and stop the following iteration if the stopping criterion is met.*
- *For* $t = 0, 1, \ldots$
  - *Generate* $\theta_s \sim q_{\phi^{(t)}}(\theta)$, $s = 1, \ldots, S$
  - *Compute the unbiased estimate of the ELBO gradient denoted :*

$$\widehat{\nabla_\phi \mathcal{LB}(\phi^{(t)})} := \frac{1}{S}\sum_{s=1}^{S} \nabla_\phi \log q_\phi(\theta_s) \times h_\phi(\theta_s)|_{\phi=\phi^{(t)}}.$$

- *Update* $\phi^{(t+1)} = \phi^{(t)} + \alpha_t \circ \widehat{\nabla_\phi \mathcal{LB}(\phi^{(t)})}$

The algorithmic parameter $S$ is referred to as the number of Monte Carlo samples and $\circ$ denotes the Hadamard product (element-wise multiplication). The sequence of learning rates $\{\alpha_t\}$ should satisfy the theoretical requirements $\alpha_t > 0$, $\sum_t \alpha_t^2 < \infty$ to converge to a local optimum (Robbins and Monro [1951]). When the variance of the gradient of the ELBO is large, the learning rate $\alpha_t$ should be small, otherwise the update $\phi^{(t+1)}$ will jump all over the place. Denote $g_t := \widehat{\nabla_\phi \mathcal{LB}(\phi^t)}$ be the gradient vector at step $t$, and $v_t := (g_t)^2$.

The commonly used adaptive learning rate methods such as ADAM (Kingma and Ba [2014]) used within this thesis (https://pytorch.org/docs/stable/generated/torch.optim.Adam.html) and AdaGrad (Duchi et al. [2011]) work by scaling the coordinates of $g_t$ by their corresponding variances. These variances are estimated by moving average following :

**Algorithm 2** (Adaptive learning methods)**.**

- *Initialize* $\phi^{(0)}, g_0, v_0$ *and set* $\bar{g} = g_0, \bar{v} = v_0$. *Let* $\beta_1, \beta_2 \in (0,1)$ *be adaptive learning weights.*
- *For* $t = 0, 1, \ldots,$ *update :*

  $\bar{g} = \beta_1 \bar{g} + (1-\beta_1) g_t$

  $\bar{v} = \beta_2 \bar{v} + (1-\beta_2) v_t$

  $\phi^{(t+1)} = \phi^{(t)} + \gamma_t \frac{\bar{g}}{\sqrt{\bar{v}}}$,

*with* $\gamma_t$ *a scalar step size.*

Stochastic gradient methods have been adapted to various settings, such as variational autoencoders ([Doersch, 2016, Rezende and Mohamed, 2015,

Kingma and Welling, 2013]) described in this Chapter. While the use of stochastic gradients substantially reduces the per-iteration cost in Algorithm 1, it introduces additional variance into the sampling process at each step, necessitating a larger total number of iterations. In practice, score function gradients often have high variance and thus are frequently used in conjunction with variance reduction techniques (Miller et al. [2017]) such as the reparameterization trick (Kingma and Ba [2014]).

### 3.1.4 Reparameterization trick

The reparameterization trick is a way to rewrite the expectation so that the distribution with respect to which we take the gradient is independent of the parameter $\theta$. Suppose that for $\theta \sim q_\phi(\cdot)$, there exists a deterministic function $g(\phi, \epsilon)$ such that $\theta = g(\phi, \epsilon) \sim q_\phi(\cdot)$ where $\epsilon \sim p_\epsilon(\cdot)$. We emphasise that $p_\epsilon(\cdot)$ must not depend on $\phi$. For example, if $q_\phi(\theta) = \mathcal{N}(\theta; \mu, \sigma^2)$, then $\theta = \mu + \sigma\epsilon$ with $\epsilon \sim \mathcal{N}(0, 1)$.

We can write the ELBO $\mathcal{LB}(\phi)$ as an expectation with respect to $p_\epsilon(\cdot)$

$$\mathcal{LB}(\phi) = \mathbb{E}_{\epsilon \sim p_\epsilon}\Big[h_\phi(g(\phi, \epsilon))\Big], \tag{3.14}$$

where $\mathbb{E}_{\epsilon \sim p_\epsilon}(\cdot)$ denotes expectation with respect to $p_\epsilon(\cdot)$.

If we differentiate under the integral sign,

$$\nabla_\phi \mathcal{LB}(\phi) = \mathbb{E}_{\epsilon \sim p_\epsilon}\Big[\nabla_\phi (g(\phi, \epsilon))^\top \nabla_\theta h_\phi(\theta)\Big] + \mathbb{E}_{\epsilon \sim p_\epsilon}\Big[\nabla_\phi h_\phi(\theta)\Big], \tag{3.15}$$

where the $\theta$ within $h_\phi(\theta)$ is understood as $\theta = g(\phi, \epsilon)$ with $\phi$ fixed so that $\mathbb{E}_{\epsilon \sim p_\epsilon}\Big[\nabla_\phi h_\phi(\theta)\Big] = 0$. Finally, the gradient in the Equation 3.15 can be estimated unbiasedly using i.i.d samples $\epsilon_S \sim p_\epsilon(\cdot), s = 1, \ldots, S$ as :

$$\widehat{\nabla_\phi \mathcal{LB}(\phi)} = \frac{1}{S} \sum_{s=1}^{S} \nabla_\phi g(\phi, \epsilon_s)^\top \nabla_\theta h_\phi(g(\phi, \epsilon_s)) \tag{3.16}$$

For each parameterization, the accuracy of the estimator in Equation 3.16

depends on the number of Monte Carlo samples $S$. With the reparameterization trick, a small $S$ is often enough to estimate the lower bound gradient. The most popular variational Bayes approach utilised throughout the thesis is Gaussian variational Bayes, where the approximation $q_\phi(\theta)$ is a Gaussian distribution with a mean $\mu$ and covariance matrix $\Sigma$.

## 3.2 Physics-informed Variational Auto Encoder

### 3.2.1 Autoencoding Variational Bayes

Variational autoencoders (VAEs) ([Kingma and Welling, 2013]) are a type of probabilistic model designed to learn latent, low-dimensional representations of data. As the name suggests, VAEs belong to the family of autoencoders [Tschannen et al., 2018, Vincent et al., 2008]. An autoencoder is a model that takes a vector $x$, compresses it through an encoder function $h_\phi(x)$ ($\phi$ a given parameter) into a lower-dimensional vector $z$, and then decompress $\boldsymbol{z}$ through a decoder function $f_\theta(z)$ ($\theta$ is a given parameter) back into $x$ with the following basic architecture :

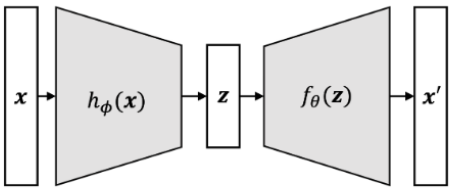



Figure 3.1: Standard autoencoder architecture. Retrieved from https://mbernste.github.io/posts/vae/ on September 3, 2023.

Autoencoders aim to minimise the reconstruction error between the input and the output of the decoder, and the learned latent representation can be used for tasks such as data compression, data visualisation, and anomaly

detection (Krissaane et al. [2019]). Standard autoencoders do not have any probabilistic assumptions and do not model any probability distribution over the input data by contrast with Variational Auto Encoder (VAE) ([Pu et al., 2016, Burda et al., 2015]) that learns a probabilistic distribution over the latent space. The VAE framework provides a method for jointly learning deep latent-variable models and corresponding inference models using stochastic gradient descent. We will introduce an encoder, detailed in Section 3.2.2, and a decoder, explained in Section 3.2.3, both connected through a loss function. This loss function is minimised to infer latent parameters of the underlying process. In the preceding section (Section 3.1), we introduced the approximate posterior $q_\phi(\theta \mid x)$. In the context of the Variational Autoencoder, this approximate posterior corresponds to the encoder. The objective is to optimize the variational parameters $\phi$ such that:

$$q_\phi(\theta) = q_\phi(\theta \mid x) \approx p(\theta \mid x). \tag{3.17}$$

The optimization objective of the variational autoencoder, like in other variational methods is the evidence lower bound, abbreviated as $\mathcal{LB}$.

$$\mathcal{LB}(\phi) = \mathbb{E}_{q_\phi}\left[\log\Big(\frac{p(\theta)p(x \mid \theta)}{q_\phi(\theta)}\Big)\right]. \tag{3.18}$$

The likelihood model $p(x \mid \theta)$ forms the decoding component, with additional details outlined in Section 3.2.3.

### 3.2.2 Variational encoder approximation

**Assumption 1.** *The variational approximation is* $q_\phi(\theta) = \mathcal{N}(\theta \mid \mu, \Sigma)$, *with* $\mu = (\mu_1, \ldots, \mu_k)^\top$ *and* $\Sigma = diag(\exp(2\lambda_1), \ldots, \exp(2\lambda_k)))$, $\lambda_i = \log(\sigma_i), \sigma_i = \Sigma_{ii}^{\frac{1}{2}}$, *where* $\mathcal{N}(\cdot \mid \mu, \Sigma)$ *denotes the Gaussian density with mean vector* $\mu$ *and (diagonal) covariance matrix* $\Sigma$.

Assumption 1 implies an independence structure known as MFVB explained in 3.1.2. These assumptions are very useful for making the optimization tractable but would be highly restrictive when the parameters are not independent in Chapter 5.

Under this assumption, the variational distribution takes the form :

$$q_\phi(\theta) = \prod_{i=1}^{k} \mathcal{N}(\theta_i \mid \mu_i, \exp(2\lambda_i)), \tag{3.19}$$

with the variational parameters $\mu = (\mu_1, \mu_2, \ldots, \mu_k)^\top$ and $\lambda = (\lambda_1, \ldots, \lambda_k)^\top$, and the vector of all variational parameters is $\phi = (\mu^\top, \exp(2\lambda)^\top)^\top$. The gradient of $\mathcal{LB}(\phi)$ is then partitioned as :

$$\nabla_\phi \mathcal{LB}(\phi) = (\nabla_\mu \mathcal{LB}(\phi)^\top, \nabla_\lambda \mathcal{LB}(\phi)^\top)^\top.$$

Following the reparametrization trick, we assume in this thesis that $\theta \sim q(\theta; \phi)$ when $q(\theta; \phi) \sim \mathcal{N}(\mu_\phi, \sigma_\phi^2)$ with $\lambda = \log(\sigma_\phi)$. Then $g(\epsilon, \theta) = \mu + \exp(\lambda) \circ \epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$ where $I$ is the $k \times k$ identity matrix. The gradient of the ELBO under reparametrization is given in Equation 3.15 and its estimate in Equation 3.16.

### 3.2.3 ODE-Informed Decoder

In this architecture, $\theta \in \mathbb{R}$ is the parameter of the first-order ODE:

$$\frac{dx}{dt} = f_\theta(x, t), \tag{3.20}$$

where $x : \mathcal{T} \to \mathbb{R}^d$ is the state function of a continuous, real-valued time variable $t \in \mathbb{R}$. The variational autoencoder architecture [Roeder et al., 2019] is employed, with the decoder being the dynamic model.

The state evolution will be governed by a first-order derivative with a known

and fixed initial value $x(t_0) \in \mathbb{R}^d$. Given the differential $f : \mathbb{R}^d \to \mathbb{R}^d$, the system follows an ordinary differential equation (ODE) model with state solutions :

$$x(T) = x(t_0) + \int_{t_0}^{T} f_\theta(x, t)dt, \tag{3.21}$$

with $\theta \in \mathbb{R}$.

The state solutions are in practice computed by solving this initial value problem with efficient numerical solvers, such as Runge-Kutta (Schober et al. [2014]). Recently, Chen's seminal paper from 2018 (Chen et al. [2018]) introduces a comprehensive framework for the differentiation of ODEs using continuous neural networks. By combining well-known sensitivity and adjoint methods (cf. Section 3.3) with modern automatic differentiation packages, the neural ODE framework (Chen et al. [2018]) enables gradient descent through ODE solvers (Massaroli et al. [2020]).

### 3.2.4 Variational Auto-encoding dynamical systems

In the section 3.2.2, the generative process is given through the Gaussian encoder using the reparametrization trick with $\lambda_\phi = \log(\sigma_\phi)$:

$$\begin{aligned} \epsilon &\sim \mathcal{N}(0, \boldsymbol{I}) \\ (\mu_\phi, \lambda_\phi) &\leftarrow \phi \\ \theta &\leftarrow \mu_\phi + \exp(\lambda_\phi) \circ \epsilon \end{aligned}$$

, where the mean-field approximation $\mathcal{Q}$ consists of distributions whose variables are all mutually independent:

$$q_\phi(\theta) = \prod_i \mathcal{N}(\theta_i \mid \mu_{\phi_i}, \exp(2\lambda_{\phi_i})). \tag{3.22}$$

The decoding process consists of solving the ODE system in Equation 3.20

with the generative process $f_\theta(x, t)$ with $\theta$ the output of the encoder and the state vector $x(t)$ over time. Solving the dynamical system with the parameter $\theta$ results in the vector $x$ assumed normal with mean the observations $y$ and a variance noise $\sigma^2$ (assumed known) :

$$x = \text{ODESolve}(f_\theta, x(t_0))$$

As seen in Section 2.3.3.3 and at the beginning of this chapter, we aim to find the variational Bayesian posterior distribution $q(\theta) = \mathcal{N}(\mu_\phi, \exp(2\lambda_\phi))$ for the latent parameter $\theta$ by maximising the ELBO $\mathcal{LB}(\phi)$ ;

$$\mathcal{LB}(\phi) = \mathbb{E}_{q(\phi)}[\log p(y \mid x, \sigma^2)] + \text{KLD}(q_\phi(\theta)||p(\theta)), \tag{3.23}$$

with $\phi$ the variational parameter to optimize. The first term is the reconstruction loss denoted as $\mathcal{L}_{\text{recon}}$. The second term denoted as $\mathcal{L}_{\text{KL}}$ can be easily computed when both distributions are Gaussians. The prior distribution $p(\theta)$ for the parameter $\theta$ is considered Gaussian, aka $p(\theta) = \mathcal{N}(\mu_{\text{prior}}, \sigma^2_{\text{prior}})$.

The complete framework is depicted in Figure 3.2. The input real data are compared to the output realizations obtained by solving the ODE with the chosen parameter $\theta$. Neural networks are not used in either the encoder or the decoder. For neural networks, the encoder processes the input data through multiple layers to produce a distribution in the latent space, typically parameterized by the mean and variance (or log-variance) of a Gaussian distribution.

We use the deep learning library PyTorch where the encoder and differential function parameters are jointly optimized using the Adam optimizer in Algorithm 1. We employ the ODE solver from the Python package `torchdiffeq` (Chen et al. [2018]) to integrate an ODE and perform back-

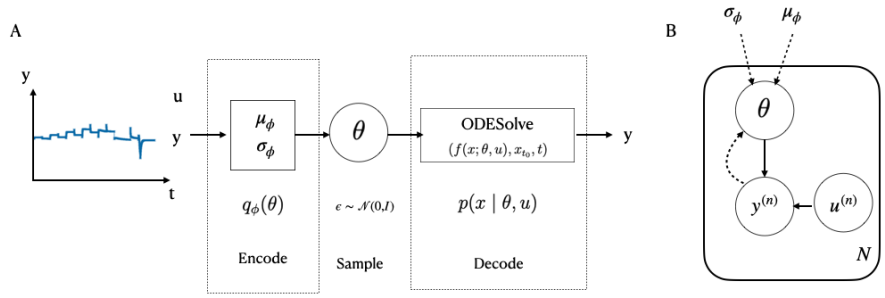



Figure 3.2: Variational Auto-encoding dynamical systems. (A): The computational flow diagram for the encoding process, sampling from the variational posterior, and simulating the dynamical system is presented. Note that the sampling and ODE solver operations are differentiable. The latent parameter of the ODE system is $\theta$ and the control parameters are given in $u$. (B): Graphical model for the trajectory variable $y$. Dashed lines represent dependencies in $q$, and solid lines in $p$. We observe data points $y$ which depend on some latent parameters $\theta$ obtained via the variational parameters $(\mu_\phi, \sigma_\phi)$.

propagation for gradient computation.

## 3.3 Automatic Differentiation

The success of Deep Learning owes much to Automatic Differentiation (AD) Automatic Differentiation (AD) (Griewank and Walther [2008]), a crucial tool enabling the efficient calculation of the gradient of useful functions with computational mathematics and computer science. Widely adopted in frameworks such as TensorFlow (Abadi et al. [2015]) and PyTorch (Paszke et al.), the key idea behind AD is to decompose all numerical computations into a finite set of elementary operations for which derivatives are known and combine the derivatives of the constituent operations through the chain rule to obtain the derivative of the overall composition.

Consider the target function $f : \mathbb{R}^n \to \mathbb{R}^m$, with $n$ independent (input) variables $x_i$ and $m$ independent (output) variables $y_j$. The corresponding $m \times n$ Jacobian matrix $J$ has $(i,j)^{th}$ component composed of the partial

**Algorithm 1** Variational Auto-encoding dynamical systems

1: **Input** : dynamic model $f_\theta(x,t)$, dataset $y_{1:n}$, initial state $x(t0)$, data noise $\sigma^2$, prior $p(\theta) \sim \mathcal{N}(\mu_{\text{prior}}, \sigma^2_{\text{prior}})$, Initialize variational parameters $\phi$ , sampling size M, learning rate $\alpha$
2: **repeat**
3: $\mu_\phi, \lambda_\phi \leftarrow$ **Encoder** $(\phi)$
4: Generate noise $\epsilon^i \sim \mathcal{N}(0, I), i = 1, .., m$
5: **for** $l \leftarrow 1$ to M **do**
6: $\theta \leftarrow \mu_\phi + \exp(\lambda_\phi) \odot \epsilon^{(l)}$ {Reparametrization trick}
7: **end for**
8: **Define the augmented ODE** {Forward/Reverse mode}
9: **Decoder** $x \leftarrow$ ODESolve $(f_\theta, x(t_0))$
10: $\mathcal{L}_{\text{recon}} \leftarrow \log p(x \mid y, \sigma^2) = -n \log(\sigma\sqrt{2\pi}) - \sum_i^n \frac{(y_i - x_i)^2}{2\sigma^2}$
11: $\mathcal{L}_{KL} \leftarrow (\log(\frac{\sigma_\phi}{\sigma_{\text{prior}}}) + \frac{\sigma_\phi^2 + (\mu_\phi - \mu_{\text{prior}})^2}{2\sigma^2_{\text{prior}}} - \frac{1}{2})$
12: $\mathcal{LB} \leftarrow \mathcal{L}_{\text{recon}} - \mathcal{L}_{KL}$ {Loss function}
13: $\phi \leftarrow \phi + \alpha \nabla_\phi \mathcal{LB}$ {Gradient descent}
14: **until** convergence of $\phi$
15: **return** $\phi$

derivative of the $j^{th}$ output with respect to the $i^{th}$ input.

$$J_{ij} = \frac{\partial y_j}{\partial x_i}$$

The ELBO $\mathcal{LB}(\phi)$ in the VAE has only one-dimensional output so the Jacobian matrix is simply the gradient vector $\nabla \mathcal{LB}(\phi)$.

When $f$ is a composite function: $f(x) = h \circ g(x) = h(g(x))$, with $x \in \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^k$ and $h : \mathbb{R}^k \to \mathbb{R}^m$. The chain rule with the elementary matrix multiplication gives :

$$J_{ij} = \frac{\partial y_i}{\partial x_j} = \frac{\partial h_i}{\partial g_1}\frac{\partial g_1}{\partial x_j} + \frac{\partial h_i}{\partial g_2}\frac{\partial g_2}{\partial x_j} + \ldots + \frac{\partial h_i}{\partial g_k}\frac{\partial g_k}{\partial x_j}.$$

More generally, if the objective function $f$ is the composite of $R$ functions, $f = f^R \circ f^{R-1} \circ \ldots \circ f^1$, the Jacobian matrix satisfies $J = J_R \cdot J_{R-1} \cdot \ldots \cdot J_1$. We further illustrate the mechanisms of AD, by considering the $\mathcal{L}_{\text{recon}}$, in line 11 of the Algorithm 1 where the log-likelihood function for an observed

data point $y$ is :

$$f(y) := \log p(y \mid \mu, \sigma^2) = -\frac{1}{2}\Big(\frac{y-\mu}{\sigma}\Big)^2 - \log(\sigma) - \frac{1}{2}\log(2\pi). \tag{3.24}$$

AD can operate in forward or reverse mode also known as adjoint mode explained below. In particular, PyTorch supports reverse-mode (Speelpenning [1980]) AD for scalar functions where Variables store extra metadata for the computation of the gradients.

### 3.3.1 Forward Mode

AD in forward mode uses intermediate variable $v_i$ with the input variable $x$ that denotes the directed partial derivative of $v$ with respect to that one root variable :

$$\dot{v}_i = \frac{\partial v_i}{\partial x} \tag{3.25}$$

Taking back the log-likelihood in Equation 3.24, we can split into elementary operations given in Figure 3.3 yielding the composite structure:

$$(y, \mu, \sigma) \leftarrow (y-\mu, \sigma) \leftarrow (\Big(\frac{y-\mu}{\sigma}\Big), \log\sigma) \leftarrow \ldots \leftarrow \log p(y \mid \mu, \sigma^2)$$

The partial derivatives of composite expressions with respect to the parameter $\mu$ are obtained with the chain rule where we set only one of the variables $\dot{\mu} = 1$ and the rest to zero :

$$\frac{\partial f(y)}{\partial \mu} = \frac{\partial f(y)}{\partial v_{10}}\frac{\partial v_{10}}{\partial v_9}\cdots\frac{\partial v_1}{\partial \mu}$$

Consider $v_5$, which is connected to $v_4$ and $v_3$ via the graph 3.3:

$$\frac{\partial v_5}{\partial \mu} = \frac{\partial v_5}{\partial v_4} \times \frac{\partial v_4}{\partial \mu} = \frac{1}{\sigma} \times (-1)$$

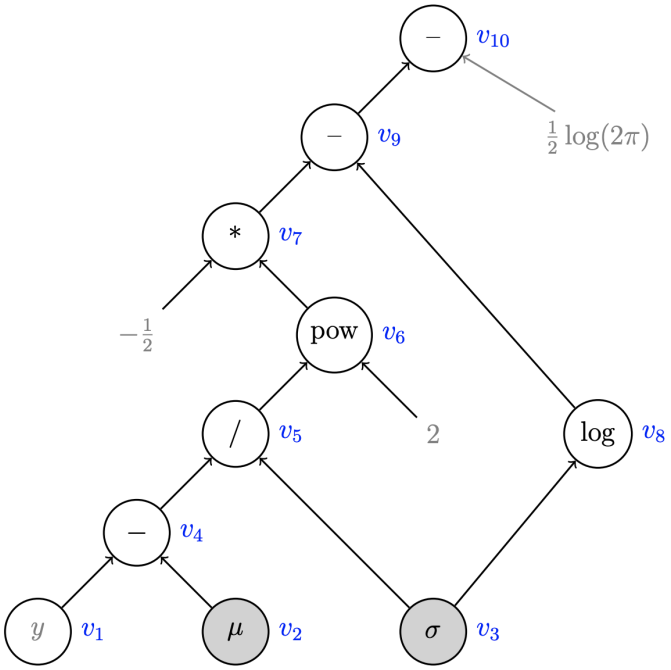



Figure 3.3: Expression graph for the log-normal density for Equation 3.24. At the top of the graph, we have the output $f(y)$ with each node being an intermediate variable and the inputs $y, \mu, \sigma$.

At each forward step, we compute one column of the Jacobian matrix that constitutes the partial derivatives of all the outputs with respect to one input. We can hence compute a full Jacobian matrix in $n$ forward steps. We know how the gradients of the ELBO composed of both the log-likelihood and the KLD can be obtained through the procedure above. However, in Algorithm 1, the ODE solver used involves steps that are automatically saved for AD with respect to $\phi$.

### 3.3.2 Reverse Mode

AD in the reverse mode propagates derivatives backward from a given output. We add to each intermediate variable $v_i$ an adjoint:

$$\bar{v}_i = \frac{\partial y_j}{\partial v_i},$$

which represents the sensitivity of a considered output $y_j$, with respect to changes in $v_i$. After executing a forward evaluation, as was done for forward mode, the reverse pass calculates the adjoints. We start with the final output, setting its adjoint with respect to itself to 1, and compute successive adjoints until we reach the initial variables of interest. One row of the Jacobian matrix is obtained at each reverse step, corresponding to the adjoints of all the inputs with respect to one output. For the ELBO objective function, with only one-dimensional output, this row corresponds exactly to the gradient.

In terms of complexity, the forward mode is on the order of $\mathcal{O}(n)$, where $n$ is the dimension of the input vector while the reverse mode is on the order of $\mathcal{O}(m)$ with $m$ is the dimension of the output vector. Since most loss functions in Machine Learning including the ELBO are scalar, the reverse mode is more favorable, which explains the popularity of back-propagation AD. However, the reverse mode implies going through the expression graph twice: one forward trace to get the value of the function and the intermediate variables; and one reverse trace to get the gradient. This means performance overhead because the expression graph and the values of the intermediate variables need to be stored in memory, especially with the use of an ODE solver through differentiation.

### 3.3.3 Differentiation through ODE solvers

Considering the dynamics in Equation 3.20 with the parametric function $f_\theta(x, t)$ and $x \in \mathbb{R}^d$, we define the sensitivity of the state vector $x$ with respect to model parameter $\theta$ at time $t$ by :

$$s(t; \theta) = \frac{dx(t)}{d\theta}. \tag{3.26}$$

The forward method will extend the ODE system in Equation 3.20 with the

Jacobian $\frac{df}{dx}$ of the derivative function $f$ with respect to the current state $x_t$, and the gradient of the derivative function $\frac{df}{d\theta}$ with respect to the parameter $\theta$. The forward-sensitivity approach works by deriving a differential equation for $s(t;\theta)$:

$$\begin{cases} \frac{ds}{dt}(t,\theta) = \frac{d}{dt}\frac{dx}{d\theta} \\ \qquad\qquad = \frac{d}{d\theta}\frac{dx}{dt} \text{ by Clairaut's theorem} \\ \qquad\qquad = \frac{d}{d\theta}f(x;\theta) \\ \qquad\qquad = \sum_i \frac{\partial f}{\partial x_i}\frac{dx_i}{d\theta} + \frac{\partial f}{\partial \theta} \text{ by the chain rule} \\ \qquad\qquad = \sum_i \frac{\partial f}{\partial x_i}s_i + \frac{\partial f}{\partial \theta}. \end{cases} \tag{3.27}$$

By solving this enlarged system, for each parameter, an additional full set of sensitivity states is added. Given $d$ state equations and $m$ parameters, the total size of the ODE system is $\mathcal{O}(d + d \cdot m)$.

The adjoint ODE solver method differs from the forward ODE solvers since it integrates forward in time a system of $d$ equations to compute the ODE solution and then integrates backwards in time another system of $d$ equations to get the sensitivities bringing the size to $\mathcal{O}(d + d + m)$. For any scalar loss function $L$ that depends on the output of the ODE solver (which is the case for the ELBO),

$$L(x_T) = L\Big(\int_{t_0}^{T} f_\theta(x,t)dt\Big), \tag{3.28}$$

where $f_\theta(x,t)$ describes the dynamics in Equation 3.20.

Pontryagin [1962] shows that its derivative takes the form of another initial value problem :

$$\frac{dL}{d\theta} = -\int_{t_0}^{T}\Big(\frac{\partial L}{\partial x(t)}\Big)^T \frac{\partial f_\theta(x,t)}{\partial \theta}dt. \tag{3.29}$$

The quantity $a(t) = -\frac{\partial L}{\partial x(t)}$ is the adjoint state of the ODE. The adjoint

method ([Chen et al., 2018, Cun, 1988]) provides :

$$\frac{da(t)}{dt} = -a^T(t)\frac{\partial f_\theta(x,t)}{\partial x}. \tag{3.30}$$

We then combine Equation 3.30 with the ODE in Equation 3.20 and solve backward-in-time the adjoint equation within reverse-mode automatic differentiation.

**Example 7** (Pendulum system)**.** *The differential equation that represents the motion of a simple pendulum where $g$ is the magnitude of the gravitational field, $\ell$ is the length of the cord, and $x$ is the angle from the vertical to the pendulum is :*

$$\frac{d^2x_t}{d^2t} + \frac{g}{\ell}\sin(x_t) = 0 \tag{3.31}$$

*Consider the pendulum system :*

$$\frac{d^2x_t}{dt^2} = -\theta\sin(x_t) := f_\theta(x(t))$$

*with $x(0) = 1$ and $\dot{x}(0) = 0$. We can rewrite the previous equation as a two-dimensional model with $\theta$ as the parameter of interest.*

$$\frac{dx_1}{dt} = x_2 \quad \frac{dx_2}{dt} = -\theta\sin(x_1), \tag{3.32}$$

*where $x_1$ and $x_2$ both are states functions of the time $t$ and*

$$f_\theta(x_t) = \begin{pmatrix} x_2 \\ -\theta\sin(x_1) \end{pmatrix}.$$

*The sensitivity is given by:*

$$\begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} \frac{dx_1}{d\theta} \\ \frac{dx_2}{d\theta} \end{pmatrix}.$$

*where we obtain the ODE gradients via :*

$$\frac{\partial f}{\partial \theta} = \begin{pmatrix} 0 \\ -\sin(x_1) \end{pmatrix},$$
$$\frac{\partial f}{\partial x_1} = \begin{pmatrix} 0 \\ -\theta\cos(x_1) \end{pmatrix},$$
$$\frac{\partial f}{\partial x_2} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

*Putting this together, we have the four-dimensional ODE given by :*

$$\frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ -\theta\sin(x_1) \\ s_2 \\ -a\cos(x_1)s_1 - \sin(x_1) \end{pmatrix}$$

*with initial conditions* $x_1(0) = 1, x_2(0) = 0, s_1(0) = 0, s_2(0) = 0$.

*In the ELBO term* $\mathcal{LB}(\phi)$ *given in Equation 3.7, the forward evaluation of the gradient of the expectation term* $\nabla_\theta \log p(y \mid \theta)$, *requires solving the augmented ODE system with the sensitivity terms. If* $x_i(t_i;\theta)$ *is the decoding data with the parameter* $\theta$, *we have*

$$\log p(y|\theta) = -\frac{1}{2\sigma^2}\sum(y_i - x_i(t_i;\theta))^2 + constant,$$

*which has derivative*

$$\frac{d\log p(y|\theta)}{d\theta} = \frac{1}{\sigma^2}\sum_i (y_i - x_i(t_i;\theta))\frac{dx_i(t_i;\theta)}{d\theta}.$$

*But we want the derivative of* $\log p(y \mid \theta = \mu_\phi + \exp(\lambda_\phi)\epsilon)$ *with respect to*

$\phi = (\phi_0, \phi_1) = (\mu_\phi, \log(\sigma_\phi^2))$.

$$\nabla_\phi \log p(y \mid \theta) = \nabla_\theta \log p(y \mid \theta) \begin{pmatrix} 1 \\ \frac{\exp(\phi_1/2)e}{2} \end{pmatrix}. \tag{3.33}$$

So we find that

$$\nabla_\phi \left( \frac{1}{S} \sum_{s=1} p(y \mid \mu_\phi + \exp(\lambda_\phi) e_s) \right) = \frac{1}{S} \sum_{s=1}^{S} \frac{d \log p(y \mid \theta_s)}{d\theta_s} \begin{pmatrix} 1 \\ \frac{exp(\phi_1/2)e_s}{2} \end{pmatrix},$$

where $\theta_s = \mu + \tau e_s = \phi_0 + \exp(\phi_1/2)e_s$.

## 3.4 Mechanistic misspecified models

### 3.4.1 Example 1: Free fall with air resistance

For the true model, consider an object in free fall near the surface of the earth. This is a two-dimensional system described by a displacement velocity state vector $(x, y)$. Assuming the object has constant acceleration $\theta$ and is subject to Stokes'drag with coefficient $\delta$, the differential equations determining the true system with respect to time are :

$$\frac{dy}{dt} = -\theta - \delta y \quad \frac{dx}{dt} = y \quad \textit{(True dGp)}. \tag{3.34}$$

For our misspecified model, we will omit the air resistance and assume a constant acceleration $\theta$. This is equivalent to the model:

$$\frac{d^2x}{d^2t} = -\theta \quad \textit{(Misspecified model)}, \tag{3.35}$$

where $\theta$ is the parameter of interest.

Figure 3.4 shows how the discrepancy increased with both times and increasing parameter $\delta$. This example represented the Type 2 misspecification described in Chapter 2.

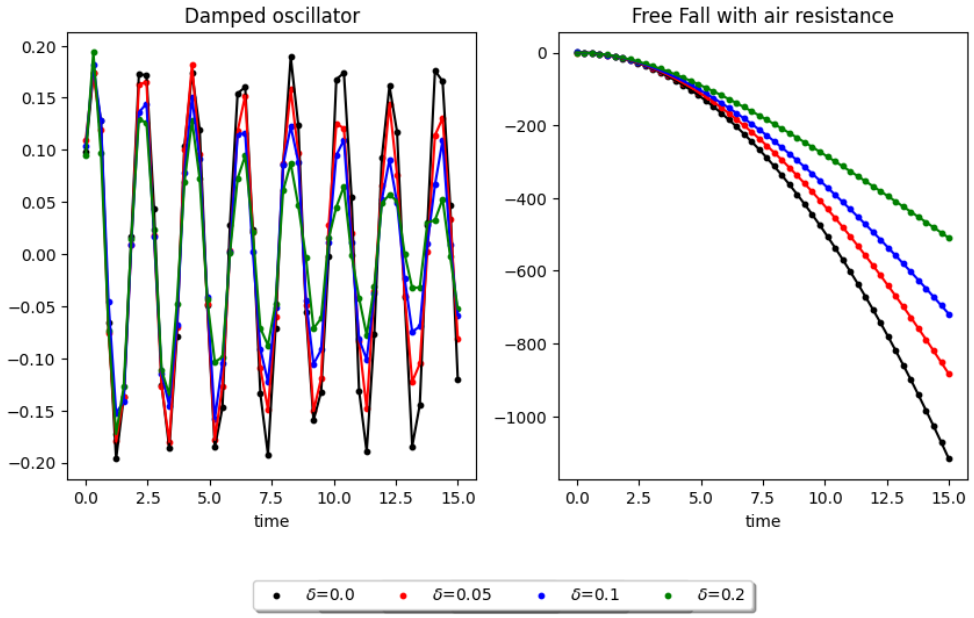



Figure 3.4: State trajectories for Equation 3.34 and 3.38 with an initial condition of $x_0 = (0.1, 0.5)$, parameter $\theta = 10$, and data noise $\sigma^2 = 0.01$ (left plot) and $\sigma^2 = 0.1$ (right plot), are depicted with varying values for $\delta$.

### 3.4.2 Example 2a: Simple gravity pendulum with small angle approximation

The differential equation that represents the motion of a simple pendulum where g is the magnitude of the gravitational field, $\ell$ is the length of the cord, and $x$ is the angle from the vertical to the pendulum is :

$$\frac{d^2x}{d^2t} + \frac{g}{\ell}\sin(x) = 0 \quad \textit{(True dGp)}. \tag{3.36}$$

We choose a misspecified model with a small angle approximation, i.e $\sin(x) \approx x$ if $x << 1$ :

$$\frac{d^2x}{d^2t} = -\theta x \quad \textit{(Misspecified model)} \tag{3.37}$$

where $\theta$ is the parameter of interest. This aligns with Type 1 misspecification, where the number of parameters is valid, but the model is not capturing some underlying dynamics.

### 3.4.3 Example 2b: Simple harmonic motion with air resistance

Consider a harmonic oscillator, which is characterised by an equilibrium position and a restoring force proportional to the displacement with friction. The system state will be a pair $(x, y)$ representing position and momentum. The change in the system with respect to time is given by the following differential equations :

$$\frac{dy}{dt} = -\theta x - \delta y \quad \frac{dx}{dt} = y \quad \textit{(True dGp)} \tag{3.38}$$

The misspecified model will omit the air resistance and assume a constant acceleration $\theta$ :

$$\frac{dx}{dt} = y \quad \frac{dy}{dt} = -\theta x \quad \textit{(Misspecified model)} \tag{3.39}$$

If $\delta = 0$, we recover the true dGp and the model is similar to the previous example 3.4.2 without a sinusoidal term. This last example represents the Type 2 misspecification and we can observe different state trajectories in Figure 3.4 where the amplitude differs with the value of $\delta$.

## 3.5 Methodology

**KL Variational Objective**: The variational Bayesian posterior distribution $q(\theta) = \mathcal{N}(\mu_\phi, \exp(2\lambda_\phi))$ for the one-dimensional latent parameter $\theta$ is obtained by maximising the ELBO $\mathcal{LB}(\phi)$ ;

$$\mathcal{LB}(\phi) = \mathbb{E}_{q(\phi)}[\log p(y \mid x, \sigma^2)] + \mathrm{KLD}(q_\phi(\theta)||p(\theta)) \tag{3.40}$$

with $\phi$ the variational parameter to optimize, $\sigma^2$ the known data noise and $p(\theta)$ the prior on $\theta$.

**PyTorch**: We implemented our framework (Figure 3.2) using the PyTorch differentiable ODE solvers available in torchdiffeq (Chen et al. [2018]). More precisely, all ordinary differential equations (ODEs) given by the misspecified model from the previous section are resolved using the solver `dopri5` implemented within this particular package.

**Dataset**: We generate data from the true dGp:

$$\frac{d^2x}{dt^2} = f_\theta(x,t), \quad t \in [0,T],$$

with the ODE parameter $\theta \in \mathbb{R}$. Observations are modeled with errors by:

$$y_i = x_i + \epsilon_i, \quad \epsilon_i \sim^{iid} \mathcal{N}(0,\sigma^2).$$

The model $f_\theta(x,t)$ is defined with the true dGp, as defined in the previous section, for each example with different data sizes, denoted as $n = \{20, 50, 100, 150, 200\}$. The initial condition is set as $x_0 = [0.1, 0.5]$, and the discrete time values are defined as $t_0 = 0, \cdots, t_n = 10$. For the free-fall model (Example 3.4.1), $\sigma^2 = 0.1$. In both the pendulum and harmonic motion (Example 3.4.3 and 3.4.2), $\sigma^2 = 0.01$. The `dopri5` solver is used. The true parameter is $\theta = 10$ in all the examples.

For Type 2 misspecification, the true dGp is defined with a level of error denoted as $\delta$ so that:

$$\frac{d^2x}{dt^2} = f_\theta(x,t) + \delta m(x)$$

where $f_\theta(x,t)$ represents the model fitted to the data within the Variational Auto Encoding framework, as explained in Section 3.2.4 and $m(x)$ is the misspecified dynamic independent of $\theta$. For Example 3.4.1 and Example 3.4.3, $m(x) = -\frac{dx}{dt}$.

When $\delta = 0$, the model fitted to the data is well-specified corresponding to the results in Section 3.7.2.1. We generate data with different values of $\delta$, specifically $\delta = \{0.05, 0.1, 0.2\}$, and fit them against the ODE model without $m(x)$.

**Settings**: The variational inference is executed over 100 epochs with a learning rate of 0.4, utilising the Adam optimizer. A sample size of $M = 15$ is employed for the Monte Carlo estimator (refer to Algorithm 1). The prior distribution for $\theta$ is consistently set as a Gaussian distribution with a mean of $\mu_{\text{prior}} = 0$ and a variance of $\sigma_{\text{prior}} = 1$.

**MCMC**: We compare our results with random-walk Metropolis–Hastings (Hastings [1970]). Given the likelihood function $p(x \mid \theta)$, we construct a Markov chain of the model parameters $\theta^{[0]}, \theta^{[1]}, \ldots$ proposing a value of $\theta$, $\theta^*$, for the $t$-th realisation from a suitable proposal density, $q(\theta^*, \theta^{[t-1]})$. Accepting this proposal with probability

$$\min\left\{1, \frac{p(\theta^* \mid x)q(\theta^{[t-1]} \mid \theta^*)}{p(\theta^{[t-1]} \mid x)q(\theta^* \mid \theta^{t-1})}\right\}$$

yields a transition kernel that guarantees the Markov chain will converge to a stationary distribution that coincides with $p(\theta \mid x)$.

**Robustness measure**: Root mean square error (RMSE) of trajectories is used to evaluate the inference performance.

$$RMSE = \sqrt{\sum_{i}^{n} \frac{(y_i - \hat{y}_i)^2}{n}},$$

where $y_i$ represents the noisy data from the dGp, $n$ denotes the data size, and $\hat{y}_i$ is the predicted trajectory obtained by solving the ODE system of

the misspecified model with the variational posterior mean $\mu_\phi$:

$$\frac{dx}{dt} = f_{\mu_\phi}(x, t).$$

## 3.6 Computational challenges

Automatic differentiation can lead to high computational demands, particularly in the context of solving ODE systems. The memory usage becomes significant in forward mode, especially when the ODE solver involves a large number of steps.

The most memory-intensive operation is the single backward call in the VAE with `torchdiffeq` (Chen et al. [2018]). This is because backpropagation through `odeint` involves extensive computation. When the adjoint method is employed, the ODE solver operates as a black box, computing gradients without backpropagating through the solver's operations, as opposed to the default `odeint`. By not storing intermediate quantities of the forward pass, we can train our models with smaller memory cost, as illustrated in Table 3.1. The adjoint sensitivity method proves effective in reducing memory requirements to $\mathcal{O}(1)$. This posed challenging computational issues, where the use of an adaptive solver was needed for most of the ODE examples considered in this thesis. For more complex ODE models, the memory usage proved to be excessively high on the virtual machine (Supermicro 620U hardware, including 32 Intel Xeon (Ice Lake class) CPU cores, 128GB of memory, and an Nvidia RTX A6000 GPU-based computations) making it impractical to scale. Tackling the challenge in dynamic models continues to be crucial, with ongoing research aimed at achieving faster model training and inference with ODEs. This exploration includes work within PyTorch ([Lienen and Günnemann, 2022, Poli and Massaroli]), as well as consideration of alternative frameworks such as Julia (Blondel

Table 3.1: Execution Times in seconds (unless stated otherwise) of ODE Solvers for Default and Adjoint Methods on the Pendulum Model (Example 3.4.3). The bold items correspond to the ODE solver employed in this chapter.

| Differentiable ODE Solvers within torchdiffeq | | |
|---|---|---|
| ODE solvers | Default odeint | odeint.adjoint |
| `Euler` | **15.5** | **88 ms** |
| `midpoint` | 26 | 158 ms |
| `rk4` | 61 | 273 ms |
| `explicit_adams` | 36 | 216 ms |
| `implicit_adams` | 76 | 394 ms |
| `dopri8` | 21 | 801 ms |
| `dopri5` | **31** | **1** |
| `bosh3` | 108 | 5 |
| `adaptive_heun` | NA | 88 |

et al. [2021]) and Jax ([Bradbury et al., 2018, Abeyasinghe et al., 2018]) that may offer improved capabilities.

Moreover, excluding the ODE solver part, the remaining VAE architecture uses automatic differentiation with reverse mode, leading to memory complexity that scales linearly with the number of algorithm iterations. Table 3.2 shows the computational time for the overall Variational Autoencoder Optimization with different data sizes and varying degrees of misspecification for the pendulum example (Example 3.4.3). When $\delta = 0$, the model is correctly specified, resulting in significantly reduced execution times. The execution times experience a substantial increase with larger data sizes and higher levels of misspecification errors.

Switching to forward-mode autodiff would be advantageous, especially considering the small number of parameters in the studied examples, as it exhibits time complexity that scales linearly with the number of variables. However, PyTorch currently lacks a straightforward framework for forward-mode autodiff (except a beta version where implementing forward-mode autodiff was not feasible for our specific situation), and due to time con-

Table 3.2: Variational Autoencoder optimization time in seconds for Example 3.4.3 with varying data size $n$ and error level $\delta$.

| $\delta$ \ $n$ | 20 | 50 | 100 | 150 | 200 |
|---|---|---|---|---|---|
| 0 | 30 | 207 | 630 | 1316 | 2272 |
| 0.05 | 61 | 291 | 793 | 1530 | 2587 |
| 0.1 | 95 | 372 | 957 | 1737 | 2898 |
| 0.2 | 127 | 457 | 1093 | 1955 | 3213 |

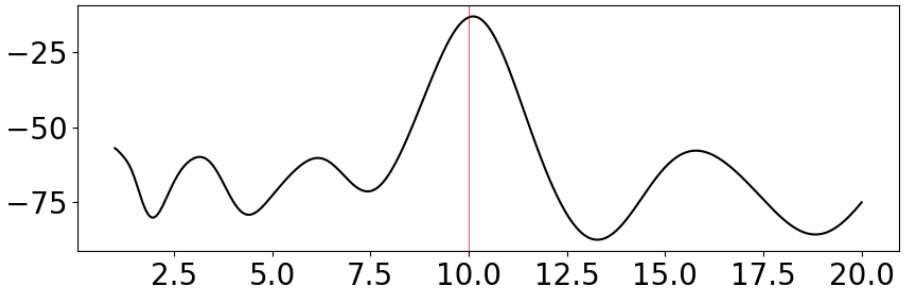



Figure 3.5: Negative Log-likelihood loss for the model in Equation 3.41 (up to proportionality) for a given parameter $\theta$. The parameter that generates the data is $\theta = 10$ represented by the vertical red line.

straints, transitioning to alternative frameworks such as CVODES (Serban and Hindmarsh [2008]) has not been possible.

## 3.7 Results

In this section, we discuss the results from our framework to learn dynamical parameters using noisy experimental measurements obtained from ODEs in well-specified and misspecified situations.

### 3.7.1 Simple Study case

Consider the simple pendulum system described in Model 2a:

$$\frac{dx}{dt} = y \quad \frac{dy}{dt} = -\theta \sin(x) \quad \textit{(True dGp)} \tag{3.41}$$

with $\theta \in \mathbb{R}$ and the state functions $(x, y)$. The initial conditions will be set up to $x(0) = 1$ and $y(0) = 0$. We simulate a trajectory of $n$ noisy observations $y$ when $\theta = 10$ with a known noise $\sigma^2 = 0.01$. The negative log-likelihood loss of this data against the ODE system displays multiple modes or peaks in Figure 3.5. This multimodality introduces complexity to optimization procedures, raising challenges for standard optimization routines that must navigate multiple minima. Therefore, if we maximise the ELBO $\mathcal{LB}(\phi)$ with respect to the parameter $\phi = (\mu_\phi, \log \sigma_\phi^2)$, we obtain the closest approximate distribution $q(\theta)$ for $\theta$ to the true posterior $p(x|\theta)$ in terms of KLD in Figure 3.6. We obtain a clear minimum value for $\mu_\phi = 10$ but a wide range of possibility for $\log \sigma_\phi^2$. This could be detrimental to uncertainty quantification because maximising the ELBO can occur across a broad range of variances. Modifying the scale unveils additional local variations. By narrowing our focus to the range $\mu_\phi \in [5, 8]$, we observe a local maximum at $\mu_\phi = 6.5$ corresponding to the optimum in that region as seen previously in Figure 3.5. The optimizer could potentially become trapped. When using MCMC with a random walk sampler, we may find the perfect posterior distribution or end up local optimum visible in Figure 3.7. The trace plots exhibit favorable characteristics for both chains, suggesting the possibility of obtaining a satisfactory approximation to the posterior distribution. However, it is crucial to acknowledge that our inference process often converges towards the mode closest to the initial chain conditions. While this tendency might be discernible in one-dimensional problems, the challenge intensifies in higher-dimensional scenarios, where potential discrepancies between the true posterior distribution and the mode captured by the chains may go unnoticed.

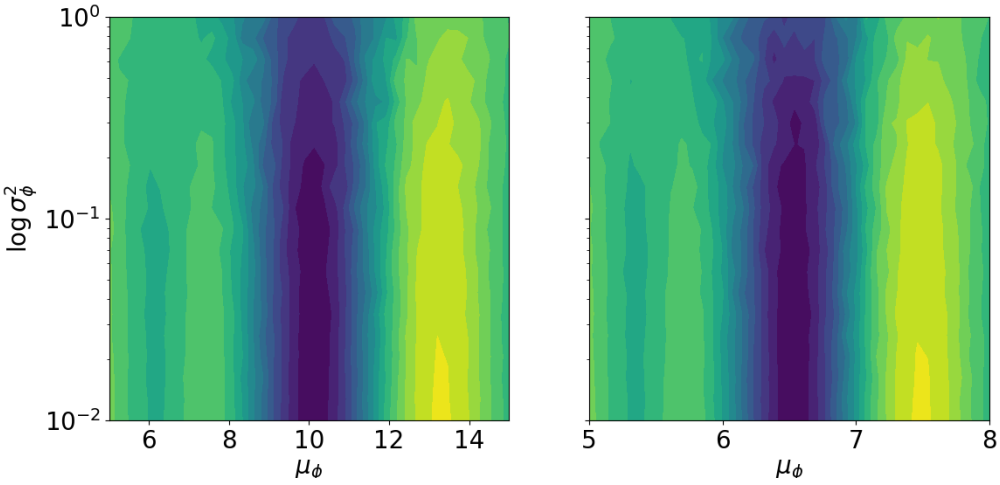



Figure 3.6: Contour plot of the negative ELBO $\mathcal{LB}(\phi)$ over possible values for $\phi = (\mu_\phi, \log \sigma^2_\phi)$. with data $x$ simulated from the pendulum system Equation 3.41) with $\theta = 10$. The darkest color on the scale denotes the maximum value for the Evidence Lower Bound (ELBO). In the right plot, the maximum value for the x-axis grid is below the true value of 10, enabling us to observe a local minimum of 6.5.

## 3.7.2 Posterior inference

### 3.7.2.1 Well-specified models

For a one-dimensional parameter, the VAE presented in Section 3.2.4 produces a posterior mean estimate $\mu_\phi$, and a posterior variance estimate $\sigma^2_\phi$ so that we obtain the closest approximate distribution $q(\theta) \sim \mathcal{N}(\theta; \mu, \sigma^2)$ for the parameter $\theta$ to the true posterior $p(x|\theta)$ in terms of KLD (cf. Equation 3.1). When the model is accurately specified, the data are fitted against the same model that generated them. Consequently, one should expect to recover a posterior distribution that closely approximates the true value $\theta$ used to simulate the data. When considering the free-fall (without air resistance) described in Equation 3.34 with $\delta = 0$, we can observe in Figure 3.8a an accurate estimation of the value $\theta$ across various data sizes. A similar result with larger variance is obtained with the simple harmonic motion (Equation 3.4.3) with $\delta = 0$, in Figure 3.8b. Surprisingly, the variational posterior variance does not notably decrease with the dataset

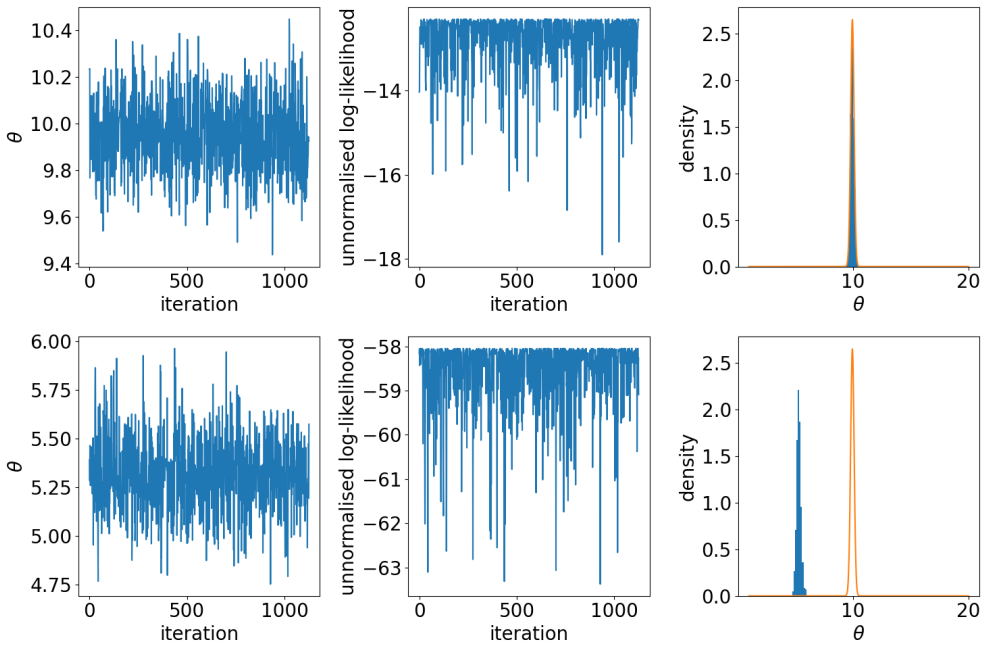



Figure 3.7: Correct and incorrect convergence of MCMC chain to global minimum. In the first column, we present two distinct MCMC trajectories obtained after removing the burn-in period (e.g., initial 500 iterations) generated using the random-walk Metropolis-Hastings algorithm. The second column shows the log-likelihood values after 500 iterations. The third column displays the density and histogram of the posterior distribution for $\theta$. In the upper plots, we observe that the chain successfully finds the true posterior distribution ($\theta = 10$). However, in the lower plots, we notice that the chain gets stuck in a local minimum.

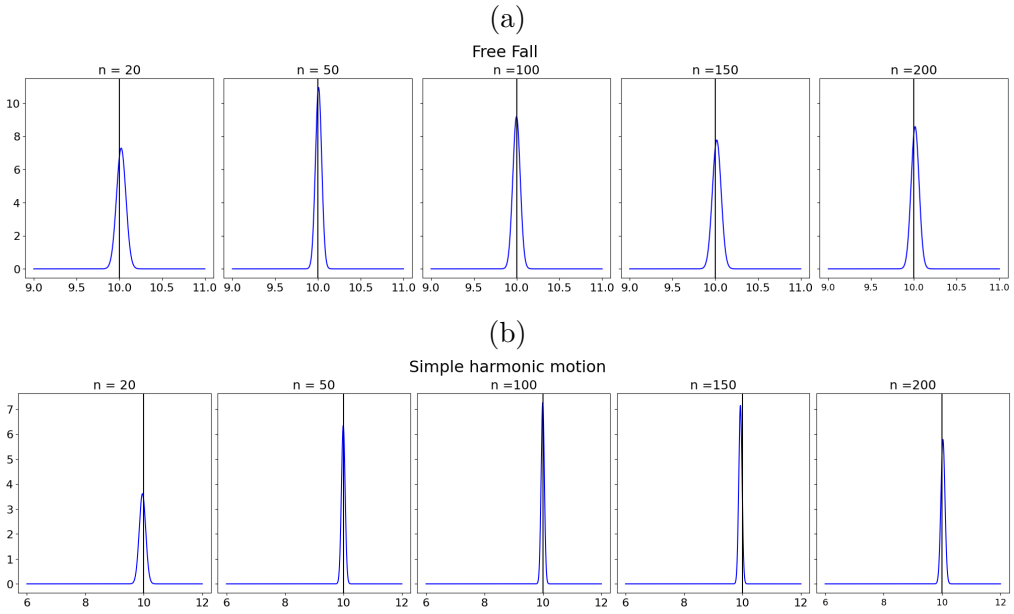



Figure 3.8: Well-specified models - Variational posterior distribution given by $q(\theta) = \mathcal{N}(\mu_\phi, \sigma_\phi^2))$ without misspecification for (a) Model 3.4.1 and (b) Model 3.4.3. The horizontal line represents the ground truth.

size (Table A.1). By deliberately omitting the initial condition from the dataset, the variational posterior for these two examples is shown in Figure A.3. Finding the posterior distribution for the pendulum example becomes challenging in this scenario. Figures A.1b illustrate the challenge of inferring $\theta$ in the presence of periodic dynamics.

#### 3.7.2.2 Misspecified models

We utilise the data generated from the dGp in the previous section, but the generative ODE model within the VAE is now incorrect. The variational posterior obtained for the three misspecified models, as detailed in Section 3.4, is presented. As a reminder, we classify them:

1. (Type I) involves using a slightly modified model based on the dGp, with the correct number of parameters (Example 3.4.2).

2. (Type II) describes a scenario where crucial parameters are omitted from the true model (Examples 3.4.1 and 3.4.3).

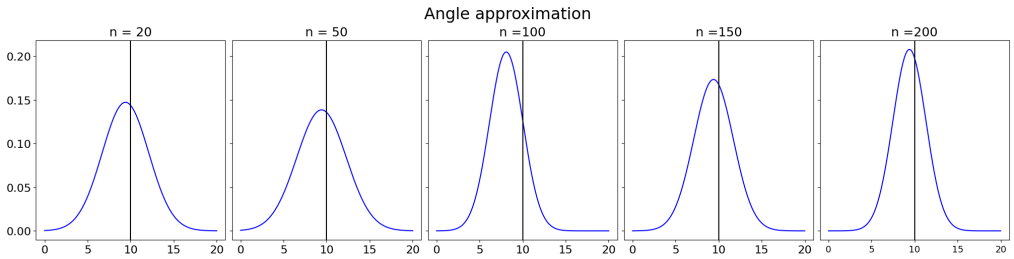



Figure 3.9: Variational posterior distribution for the misspecified small angle approximation model (Equation 3.4.2). The horizontal line represents the ground truth.

### Type I

When the model is slightly misspecified, as in the case of the pendulum example 3.4.2 where the model is missing the sinusoidal function. The variational approach accurately estimates the parameter, as depicted in Figure 3.9, irrespective of the data size. The high variational variance obtained across all data sizes leads to significant uncertainty, encompassing the true dynamics but exhibiting high variability.

### Type II

An inaccurate estimate of $\theta$ is obtained in the free-fall model shown in Figure 3.10a, 3.10c, 3.10e where the bias increases with the level of misspecification denoted by $\delta$. The average variance is small ($\sigma_\phi^2 = 0.003$), contributing to highly misleading conclusions in the inference process. We can use the approximate posterior distribution to predict the dynamical trajectory by replacing $\theta$ with $\hat{\mu}_\phi$ in the misspecified model :

$$\frac{d^2x}{d^2t} = -\hat{\mu}_\phi \quad \textit{(Misspecified model)}, \tag{3.42}$$

and compare them to the ground truth, i.e. the noisy data points. The trajectory obtained for the misspecified model is displayed alongside the

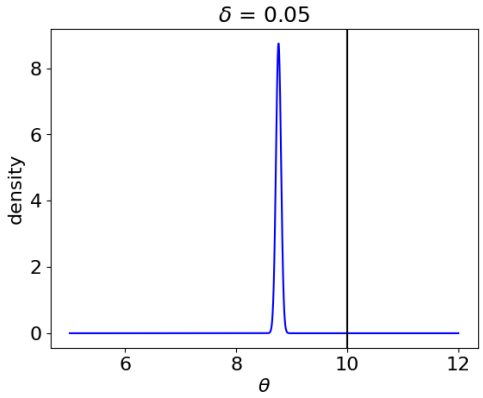
| | $\delta = 0$ | $\delta = 0.05$ | $\delta = 0.1$ | $\delta = 0.2$ |
|---|---|---|---|---|
| Free fall | 5.9 | 23.1 | 46.1 | 80.92 |
| Simple harmonic motion | 0.01 | 0.02 | 0.03 | 0.05 |

Table 3.3: Root Mean Square Error between original trajectories and reconstructed ones from the variational parameter $\mu_\phi$ for the misspecified models of Type II.

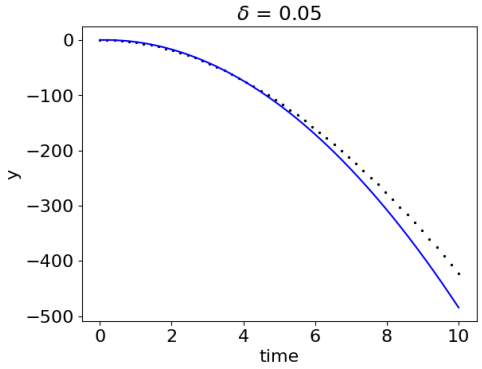
noisy data in Figure 3.10b, 3.10d, 3.10f. Despite the biased estimate, for $\delta = 0.05$, the obtained trajectory is relatively accurate, showing a small RMSE in Figure 3.10b. As the model becomes more inaccurate with increasing $\delta$, the predicted trajectories deviate further from the ground truth, revealing an interesting proportionality between the model error and the RMSE in Table 3.3. The simple harmonic motion exhibits analogous findings, with the posterior consistently underestimating the true value for $\theta$ (Table A.2). Despite this, the RMSE between the trajectory derived from the variational estimate and the true trajectory remains small in Table 3.3. This result can be explained by the periodic nature of the model, where a minor deviation in the dynamical process does not sufficiently impact the log score to update the parameter $\theta$ in the optimization process.

## 3.8 Discussion

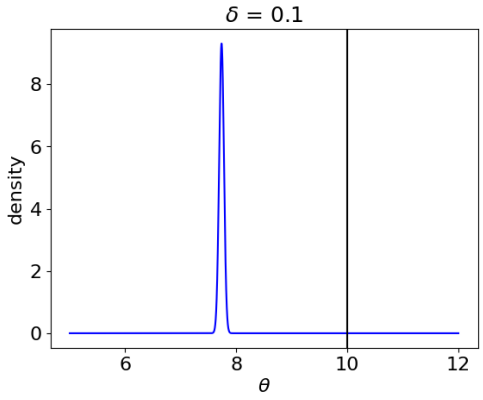
We have introduced a flexible variational Bayesian framework for the interpretable modelling of a continuous-time latent dynamical process. Our decision to employ this framework stems from leveraging recent advancements in variational inference to facilitate Bayesian parameter inference, primarily to address the practical challenges posed by Markov Chain Monte Carlo (MCMC) algorithms when applied to Ordinary Differential Equation (ODE) models. By adopting a variational autoencoder approach, we integrate mechanistic models into the generative process, specifically within



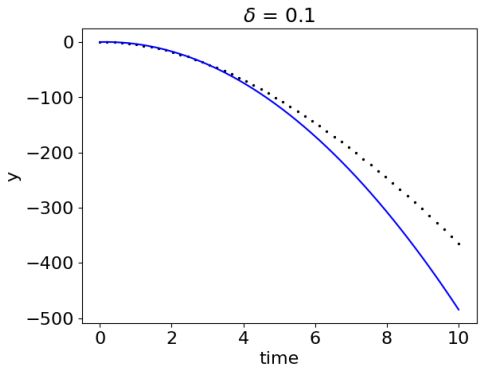

(a) $\hat{\mu}_\phi = 8.76$



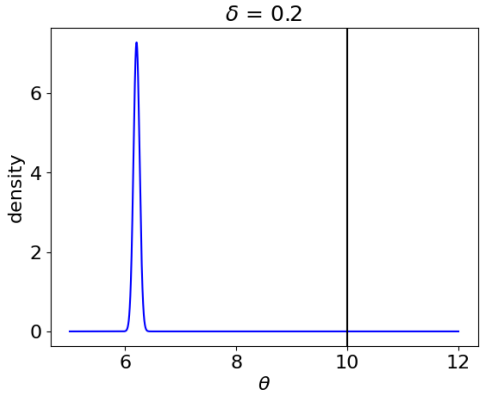

(b) RMSE = 23.1



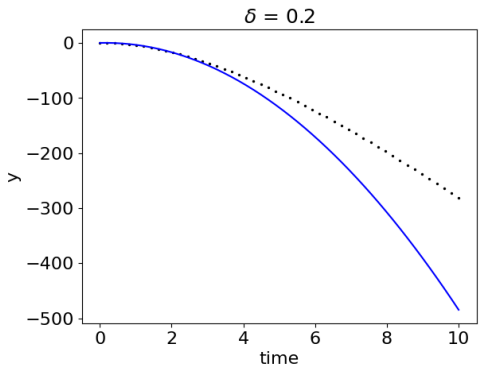

(c) $\hat{\mu}_\phi = 7.73$




(d) RMSE = 46.1




(e) $\hat{\mu}_\phi = 6.20$




(f) RMSE = 80.9

Figure 3.10: Left Panel: Variational Posterior Obtained with VAE Across Varying Error Levels $\delta$ (Top to Bottom). Right Panel: Trajectories Obtained with VAE against the observations.

the decoder component. We have simplified the conventional classical VAE framework by excluding data usage in the encoder process, and instead, we utilise a simple prior that can encapsulate domain expertise.

The effectiveness of the approach is demonstrated through data simulated from several nonlinear dynamical systems that describe physical phenomena. We have successfully recovered a meaningful latent parameter and accurately inferred the dynamic model, yielding a close approximation to the true system. While we only consider one-dimensional latent parameter inference here, it is worth noting that the VAE method can be extended to any parameter dimension. This will be further explored in Chapter 5.

Our work innovatively delved into investigating the performance of the Variational Inference model in the context of dynamical model misspecification, where an incorrect model is used to represent the true unknown dGp. In these scenarios, the variational Bayes posterior consistently converges with relatively high certainty to a point mass at $\theta^*$. This convergence leads to a misleading parameter estimate in what we categorised as a misspecification of Type II, where some crucial parameters are omitted. On the other hand, when smaller errors in the model specification are present, the VAE framework is correctly estimating the parameter of interest. This approach can be generalised to various dynamical models to enhance interpretability in data-driven modelling systems but lacks scalability.

Indeed, the ODE solver employed in this work, as detailed in Section 3.6, has posed a considerable bottleneck in terms of computational cost. Some approaches investigate restricting the flexibility of the base ODE and provide simplified ODE that is often easier to integrate (Laisk et al. [2021]). Other approaches avoid the need for explicit ODE solving and approximate free-form dynamics such as gradient matching ([Gorbach et al., 2017, Macdonald and Husmeier, 2015, Niu et al., 2016, Ramsay et al., 2005, Brunton et al., 2015, Sengupta et al., 2014]). Gradient matching involves an initial

smoothing phase where time series data is interpolated. Then the ODE parameters are optimized to minimise some metric measuring the difference between the slopes of the tangents to the interpolants, and the time derivatives from the ODEs. However, these methods face challenges in easily distinguishing simultaneous estimates of structural parameters and initial conditions, leading to reduced accuracy and increased sensitivity to noise.

## 3.9 Conclusion

In the Bayesian context of inferring a posterior distribution, Variational Inference has been widely employed, offering a general method to approximate intractable posterior distributions through an optimization approach. Recent works have incorporated dynamical models utilising variational inference ([Duncker et al., 2019, Tran et al., 2017]), particularly within the framework of variational autoencoder models ([Zhao et al., 2019, Garsdal et al., 2022, Shin and Choi, 2023]). Opting for a generative model is advantageous, as it facilitates the capture of latent structures in the data through an encoder-decoder framework.

Dynamical systems governed by Ordinary Differential Equations are prevalent in various scientific and engineering domains. Therefore, many scientific applications aim to construct a learning system capable of integrating mechanistic ODE models with data-driven methods (Linial et al. [2020], Yıldız et al. [2019]). These methods are called grey-box or hybrid approaches ([Tulleken, 1993, Kristensen et al., 2004, Astudillo and Frazier, 2021]) and combine the predictive capabilities of black-box models ([Ryder et al., 2018, Massaroli et al., 2020]) with the physical interpretability inherent in physics-based models. A category of black-box models encapsulates unknown dynamics using neural networks and depends on differentiable nu-

merical integrators, which leverage the adjoint method for memory-efficient training (Kidger et al. [2020]).

Assuming well-specified models, numerous researchers have undertaken investigations into the properties of Variational Bayes posteriors across various specific Bayesian models including linear models ([You et al., 2014, Ormerod and Wand, 2012]), exponential families (Wang and Titterington [2012]) or generalised linear mixed models ([Wang and Titterington, 2005]). The consistency and asymptotic normality of the Variational Bayes posteriors have been established in such contexts ([Wang and Blei, 2017, Pati et al., 2017, Zhang and Gao, 2017]). This was evidenced in the inference obtained within our framework, demonstrating efficiency in both bias and uncertainty quantification. However, when considering misspecified models, the performance of inference becomes suboptimal. In Chapter 4, our objective is to address this challenge by introducing recent approaches to tackle model misspecification.

# Chapter 4

# Robust losses in Generalised Variational Inference for dynamical model misspecification

**Summary:** This chapter focuses on likelihood-based robust losses in the context of model misspecification. These methods were designed to include model discrepancy and to make inference more robust. We propose to use a set of robust loss functions proposed in the General Bayes and Generalised Variational Inference litterature for learning latent parameters in misspecified models. We integrate two new losses derived from Approximate Bayesian Computation and History Matching in the Generalised Vartional Inference framework. The challenges encountered in the preceding chapter motivate the use of General Bayesian updates and our objective is to gain a comprehensive understanding of the effectiveness of these approaches when applied to a variety of misspecified dynamical models.

## 4.1 Bayes' rule in the M-open world

If the likelihood model, denoted as $p(x \mid \theta)$ turns out to be inaccurately specified, the usual workflow in Bayesian statistics (Gelman et al. [2020]) would include residual analysis, in-depth exploration of descriptive statistics, and consultation with domain experts to yield a revised likelihood model that better describes the data. In essence, the conventional perspective suggests that issues stemming from model misspecification ultimately reflect deficiencies in the modelling process itself.

As we emphasised in Chapter 3, traditional Bayesian updating can be interpreted as a method that minimises the Kullback-Leibler Divergence (KLD) between the model and the data-generating process; i.e. select the model parameters that generate simulations closest to the data in terms of KLD (Walker [2013]). Within the GVI paradigm (Section 2.3.3.3), we have seen in Chapter 2 that minimising the KLD divergence is equivalent to an optimization problem using the logarithmic loss function (Definition 2.1).

As outlined in Chapter 2 and 3, assuming that our parametric model family accurately describes the true data-generating process (dGp) is often inappropriate and can result in misleading estimates that exhibit high bias and unwanted uncertainty, which may be either too large or too small. In the robustness literature, this is often demonstrated with minor data contamination, such as outliers, causing the estimated parameters to deviate towards incorrect values. This is exemplified in the widely discussed $\epsilon$ contamination problem (see Example 2), where the underestimation of the correct parameter occurs due to contamination. This is because the likelihood model is presumed to be a reasonably accurate representation of the data, and as a result, the most informative observations are those that deviate from the model fitted to the remainder of the data.

In the specific context of inference, our objective is not necessarily to

achieve a complete and exhaustive representation of the true dGp. Instead, we aim to establish reasonable beliefs regarding the inferred parameters, with the ability to quantify uncertainty (UQ). The key concept of robustness described in Chapter 2 serves as the foundation for the robust loss functions outlined in this chapter within a Generalised Variational Inference (GVI) framework. These loss functions should strive to exhibit reduced sensitivity to departures from model assumptions, ideally resulting in improved accuracy when the model is not correctly specified. We can intuitively replace the non-robust logarithmic loss, and incorporate new loss functions into the Variational Autoencoder (VAE) discussed in Chapter 3 following the strategy outlined in Section 2.3.3.3.

## 4.2 Generalising Bayesian Inference

In Generalised Bayes Inference (GBI) (Bissiri et al. [2016], Walker [2013]), we acknowledge that the model may be misspecified or incomplete. The traditional Bayesian framework relies on a model for the dGp where the likelihood function links data and quantities of interest. This approach recognises that models are simplified representations of reality and that there may be additional complexities or nuances that are not fully captured by the chosen model structure (Matsubara et al. [2021], Matsubara et al. [2022], Jewson et al. [2018]). The general Bayesian update produces a posterior distribution over some quantity of interest without relying on a full model for the observations. Here the quantity of interest is defined via a loss function, instead of a likelihood. Bissiri et al. [2016] presented a general framework for updating targeted belief distributions of this kind. Instead of restricting themselves to parameters that index a family of distribution functions, we consider general parameters whose true value minimises an expected loss for some loss function $\ell_n : \Theta \times \mathcal{X}^n \to [0, +\infty)$. The resulting

posterior distribution is precisely the Gibbs posterior, where the loss for multiple observations is defined additively, i.e., $\ell_n(\theta, x_{1:n}) = \sum_{i=1}^{n} \ell(\theta, x_i)$.

**Definition 4.1** (General Bayes posterior (Bissiri et al. [2016]))**.**
*The General Bayes posterior given prior belief $p(\theta)$ on $\Theta$ and the observations $x_{1:n}$ is given by :*

$$p_{GB}(\theta \mid x_{1:n}) \propto p(\theta) \exp(-w\ell_n(\theta, x_{1:n})) \tag{4.1}$$

*where $w > 0$ is a calibration weight (or loss scale).*

We assume throughout the thesis that this posterior distribution is proper, i.e., the loss and prior are provided such that the right-hand side in Definition 4.1 is integrable. The Gibbs posterior $p_{GB}(\theta \mid x_{1:n})$ coincides with the Bayesian posterior (Equation 1.1) when $w = 1$ and the loss function is the negative log-likelihood.

This posterior also relates to the Generalised Variational Inference approach (Knoblauch et al. [2019]) where the posterior can be rewritten as the solution to the optimization problem :

$$P(\ell_n, \mathrm{KLD}, \mathcal{Q}) = \arg\min_{q(\theta)\in\mathcal{Q}} \mathbb{E}_{\theta\sim q(\theta)}[\sum_{i=1}^{n} w\ell(\theta, x_i)] + \mathrm{KLD}(q(\theta)\|p(\theta)) \tag{4.2}$$

where $\mathcal{Q} = \{q(\theta) : \int q(\theta)d\theta = 1\}$. The proof is equivalent to the one given in Section 2.3.3.3.

In this chapter, we propose to use the GVI framework with a set of losses $\ell_n$ and with $D = \mathrm{KLD}$, the Kullback Leibler divergence. This new form of posterior will eventually address various degrees of model misspecification, especially in dynamical models since the classical Bayesian posterior does not guarantee to be optimal in the M-open world (see Chapters 2 and 3).

## 4.3 Divergence-derived robust loss functions

The log-score and Kullback-Leibler divergence are fundamental components of Bayesian inference. However, Bayesian methods also encompass a wide range of other divergences, several of which can be conveniently linked to loss functions. We introduce the well-known families of divergences and their corresponding loss function interpretations below since they have demonstrated their benefits in the context of robust inference (Jewson et al. [2018]). This list is non-exhaustive but merely contains ones we have considered and experimented with misspecified dynamical models.

Consider $(x_1, ..., x_n)$ observed data generated independently and identically with a true density $\prod_{i=1}^{n} g(x_i)$ on a sample space $\mathcal{X}$. The general inference problem involves estimating the parameter $\theta \in \Theta$ using the likelihood model $p(x \mid \theta)$. When the model is misspecified, $g(x) \notin \{p(x \mid \theta) : \theta \in \Theta\}$.

### 4.3.1 Statistical Divergences

When considering inference in the M-open world, we can choose to measure the discrepancy between two distributions (Walker [2013]); i.e statistical divergences.

**Definition 4.2** (Statistical Divergences (Eguchi [1985]))**.**
*A statistical divergence $D(g\|f)$ is a measure of discrepancy between two probability densities $f$ and $g$ with the following two properties:*

1. $D(g\|f) \geq 0$, $\forall f, g$
2. $D(g\|f) = 0$ *if and only if* $g = f$

Divergences are often asymmetrical and do not necessarily satisfy the triangle-inequality. We now introduce several well-known families of divergences for use in inference for the misspecified models considered in this thesis, as they

are widely recognised for effectively handling model misspecification (Peng and Dey [1995], Cichocki and Amari [2010]).

**Definition 4.3** (The $\alpha$-divergence ($\alpha$D)(Chernoff [1952]))**.**

*The $\alpha$-divergence is defined as :*

$$D_A^{(\alpha)}(g(x)\|f(x)) = \frac{1}{\alpha(1-\alpha)}\left\{1 - \int g(x)^\alpha f(x)^{1-\alpha} dx\right\}, \tag{4.3}$$

*where $\alpha \in \mathbb{R}\backslash\{0,1\}$.*

**Definition 4.4** (The Rényi $\alpha$-divergence (Rényi [1961]))**.**

*The Rényi $\alpha$-divergence is defined as*

$$D_{AR}^{(\alpha)}(g(x)\|f(x)) = \frac{1}{\alpha-1}\log\left\{\int g(x)^\alpha f(x)^{1-\alpha} dx\right\}, \tag{4.4}$$

*where $\alpha \in \mathbb{R}\backslash\{0,1\}$.*

Note that Rényi-$\alpha$D is an invertible function of the $\alpha$-Divergence since

$$D_{AR}^{(\alpha)}(g(x)\|f(x)) = \frac{1}{\alpha-1}\log(1-\alpha(1-\alpha)D_A^{(\alpha)}(g(x)\|f(x))$$

**Definition 4.5** (The $\beta$-divergence ($\beta D$) (Basu et al. [1998]))**.**

*The $\beta D$ is defined as :*

$$D_B^{(\beta)}(g(x)\|f(x)) = \int g(x)\frac{g(x)^{\beta-1} - f(x)^{\beta-1}}{\beta-1} - \frac{g(x)^\beta - f(x)^\beta}{\beta} dx, \tag{4.5}$$

*where $\beta \in \mathbb{R}\backslash\{0,1\}$.*

**Definition 4.6** (The $\gamma$-divergence ($\gamma$D) (Fujisawa and Eguchi [2008]))**.**

*The $\gamma D$ is defined as :*

$$D_G^{(\gamma)}(g(x)\|f(x)) = \frac{1}{\gamma(\gamma-1)}\log\frac{(\int g(x)^\gamma dx)(\int f(x)^\gamma dx)^{\gamma-1}}{(\int g(x)f(x)^{\gamma-1}dx)^\gamma}, \tag{4.6}$$

*where* $\gamma \in \mathbb{R}\backslash\{0,1\}$.

Note that the $\alpha$D can be shown to be generated from the $\beta$D by applying the transformation :

$$c_0 \int g(x)^{c_1} f(x)^{c_2} dx \to \log\Big[\Big(\int g(x)^{c_1} f(x)^{c_2} dx\Big)^{c_0}\Big]$$

to all three terms of $\beta$D.

**Remark 1** (Link with KLD)**.**
In the limit $\alpha = \beta = \gamma \to 1$, the $\alpha$D, the Rényi $\alpha$D, the $\beta$D and the $\gamma$D recover a generalised Kullback–Leibler divergence form.

*Proof.* We demonstrate this for the $\beta$D divergence.

$$\begin{aligned}
\lim_{\beta\to 1} D_B^{(\beta)}(g(x)\|f(x)) &= \lim_{\beta\to 1} \int g(x) \frac{g(x)^{\beta-1} - f(x)^{\beta-1}}{\beta - 1} - \frac{g(x)^{\beta} - f(x)^{\beta}}{\beta} dx, \\
&= \int g(x) \log(\frac{g(x)}{f(x)}) - g(x) + f(x) dx, \\
&= \mathrm{KLD}(g(x)\|f(x)) + \int f(x) - g(x) dx,
\end{aligned}$$

where we used the identity: $\lim_{\beta\to 0} \frac{p^\beta - q^\beta}{\beta} = \log(\frac{p}{q})$. □

**Remark 2** (Link with loss functions)**.**
In the previous chapters, we have seen that minimising the log-score in expectation over data is equivalent to minimising the KLD to the dGp. In practice, some of the divergences above can be interpreted in terms of loss functions (Smith and Bernardo [2008], Dawid et al. [2014]). Denote a loss function $\ell(\theta, x)$, if the variational Bayesian posterior $q^*(\theta)$ optimization takes the form:

$$\begin{aligned}
q^*(\theta) &= \arg\min_{q\in\mathcal{P}(\Theta)} D(q(\theta)\|p(\theta \mid x)) \\
&= \arg\min_{q\in\mathcal{P}(\Theta)} D\Big(q(\theta)\|p(\theta)\exp\{-l(\theta,x)\}\, Q^{-1}\Big)
\end{aligned}$$

with $Q = \int_\theta p(\theta) \exp\{-\ell(\theta, x)\} d\theta$, the inference problem conveniently translates into the minimisation of the loss function.

### 4.3.2 Robust loss functions

When dealing with model misspecification, our primary concern is to derive robust inferences that account for the inaccuracies in the model. We derive two additive likelihood-based robust losses from the $\beta$ and $\gamma$ divergences which are well-suited for integration into the variational framework. These robust likelihood-based losses are typically less statistically efficient under correct specification, but very useful under mild misspecification (Basu et al. [1998], Fujisawa and Eguchi [2008]). For instance, they assign less weight than the KLD divergence to tail observations, thereby reducing the influence of outliers and resulting in more precise estimates (Jewson et al. [2018]).

Denote

$$I_{p,a}(\theta) = \int p(y \mid \theta)^a dy.$$

The $\beta$ loss function, denoted as $\ell^\beta$, aims to minimise the $\beta$ divergence and assess the fit of likelihood parameter $\theta$ on the sample $x_{1:n}$.

$$\ell^\beta(\theta, x_{1:n}) = \sum_{i=1}^{n} \mathcal{L}_p^\beta(\theta, x_i) = \sum_{i=1}^{n} -\frac{1}{\beta - 1} p(x_i \mid \theta)^{\beta-1} + \frac{I_{p,\beta}(\theta)}{\beta}, \tag{4.7}$$

where $\beta \in \mathbb{R} \backslash \{0, 1\}$.

The $\gamma$ loss, represented as $\ell^\gamma$ and focused on minimising the $\gamma$ divergence, can be expressed as follows:

$$\ell^\gamma(\theta, x_{1:n}) = \sum_{i=1}^{n} \mathcal{L}_p^\gamma(\theta, x_i) = \sum_{i=1}^{n} -\frac{1}{\gamma - 1} p(x_i \mid \theta)^{\gamma-1} . \frac{\gamma}{I_{p,\gamma}(\theta)^{\frac{\gamma-1}{\gamma}}}, \tag{4.8}$$

where $\gamma \in \mathbb{R} \backslash \{0, 1\}$.

The proofs are presented in Appendix A.2. In this thesis, following the same framework as in Chapter 3 (Algorithm 1), we will use the analytic form for both of these losses.

As a consequence, the variational posterior distribution $q^*(\theta)$, which optimally fits the model to the data, is given by:

$$q^*(\theta) = \arg\min_{q(\theta)\in\mathcal{Q}} \mathbb{E}_{\theta\sim q(\theta)}[\sum_{i=1}^{n} w\ell(\theta, x_i)] + \mathrm{KLD}(q(\theta)\|p(\theta)), \tag{4.9}$$

where $\mathcal{Q}$ is a tractable family of distributions, $p(\theta)$ is the prior distribution on the parameter $\theta$ and $w \in [0, 1]$ some scaling parameter. By incorporating the divergence-derived robust losses above, the updated expression for the ELBO in Equation 3.40 is:

$$\mathcal{LB}(\phi) = \mathbb{E}_{q(\phi)}[-w\ell(\theta, x)] + \mathrm{KLD}(q_\phi(\theta)||p(\theta)). \tag{4.10}$$

Here, $\mathcal{LB}(\phi)$ represents the lower bound on the evidence with respect to the variational parameter $\phi$, and the term $w\ell(\theta, x)$ encapsulates a weighted loss function. The loss function $\ell$ can take on different forms, such as $\ell^\gamma$ or $\ell^\beta$, providing flexibility in modelling and addressing specific characteristics of the data.

## 4.4 Uncertainty quantification-derived losses

The Generalised Variational Inference (GVI) objective provides an alternative perspective on Bayesian inference, framing it as a more adaptable regularised optimization process with the primary goal of minimising a specific loss function over the data. In our efforts to tackle model misspecification, we detail the loss functions derived from ABC and History Matching, both of which were described in more detail in Chapter 2.

### 4.4.1 Approximate Bayesian Computation

In the context of GVI, where we embrace a broader perspective on inference, ABC operates under the assumption of an imperfect alignment between model-generated data and observed data. The ABC posterior presented in Section 2.3.3.1 takes into account a specified tolerance parameter denoted as $h$ and is articulated as follows:

$$p_{ABC}(\theta \mid x) \propto p(\theta) \int K_h(x, y) p(y \mid \theta) dy, \tag{4.11}$$

with the prior $p(\theta)$, the observations $x$, the synthetic data $y$ (coming from a model or simulator) and the chosen kernel $K_h(y, x)$ adhering to the properties outlined in Chapter 2. We combine ABC with a variational autoencoder architecture (Moreno et al. [2016], Schmon et al. [2020], Frazier [2020]) following the Chapter 3 framework.

We define the likelihood of $p(x \mid \theta)$ as the ABC likelihood

$$p_{ABC}(x \mid \theta) = \int K_h(x, y) p(y \mid \theta) dy \approx \frac{1}{S} \sum_{s=1}^{S} K_h(x, y^{(s)}),$$

where $p_{ABC}(x \mid \theta) \to p(x \mid \theta)$ as $h \to 0$.

**Proposition 1.** *The ABC posterior presented in Equation 4.11 within the framework of Generalised Variational Inference is expressed as:*

$$P(\ell_{ABC}, KLD, \mathcal{Q}) : q^*(\theta) = \arg \min_{q(\theta) \in \mathcal{Q}} \mathbb{E}_{q(\theta)}[\ell_{ABC}(\theta, x)] + KLD(q(\theta) \| p(\theta)), \tag{4.12}$$

*with :*

$$\ell_{ABC}(\theta, x) = -\log(p_{ABC}(x \mid \theta)). \tag{4.13}$$

Similarly to what has been done in Chapter 3, a natural variational lower

bound for ABC takes the form of :

$$\mathcal{LB}_{ABC}(q) = \int q(\theta) \log \int K_h(x, y) p(y \mid \theta) dy d\theta - \mathrm{KLD}(q(\theta) \| p(\theta)), \quad (4.14)$$

with $K_h(x, y)$ and $h$ the tolerance rate.

We reinterpret the kernel $K_h$ as the measurement error model for the true data. Incorporating the ABC loss into our GVI framework involves defining a tolerance rate, representing the level of error we are willing to tolerate. In contrast to the standard interpretation in ABC, this suggests that larger bandwidths $h$ could potentially enhance the robustness of inferences.

### 4.4.2 History Matching

The History matching (HM) approach described in Chapter 2 addresses model misspecification by iteratively selecting input parameters that lead to acceptable matches with observed data (Andrianakis et al. [2015]). We define a loss function incorporating an implausible measure within HM in the generalised variational inference framework.

We define the history-matching loss to be :

$$\ell_{HM}(\theta, x) = -\log \mathbb{1}_{(\mathcal{I}(\theta,x)<c)} \quad (4.15)$$

$$= \begin{cases} 0, & \mathcal{I}(\theta, x) < c \\ \infty, & otherwise \end{cases}, \quad (4.16)$$

where $\mathcal{I}(\theta, x)$ is the implausibility measure, $x$ is a data point and $\theta$ is the parameter of interest.

For example, we can choose :

$$\mathcal{I}(\theta, x) = \frac{\mathbb{E}[X] - x}{(\mathbb{V}ar[X])^{\frac{1}{2}}}, \quad (4.17)$$

where the expectation and variance are obtained with the likelihood, i.e. $X \sim p(x|\theta)$.

Denote the not ruled out yet (NROY) set by

$$\mathcal{N} = \{\theta \in \Theta : \mathcal{I}(\theta, x_{1:n}) < c\},$$

where, e.g., $\mathcal{I}(\theta, x_{1:n}) = \max_i \mathcal{I}(\theta, x_i)$ so that $\theta$ is plausible for $x_{1:n}$ if and only if $\theta$ plausible for all $x_i$. Suppose we use the history matching loss, with the Kullback-Leibler divergence and we do not restrict the space of possible posteriors, i.e., set $P_\Theta(x)$ (no family of approximating densities $\mathcal{Q}$). Then if we define the posterior by :

$$q_{HM}(\theta) = \frac{p(\theta)\mathbb{1}_{(\theta \in \mathcal{N})}}{\int_{\mathcal{N}} p(\theta) d\theta},$$

with $p(\theta)$ the prior distribution of $\theta$. We have the following proposition.

**Proposition 2.** *$q_{HM}$ is the solution to the GVI problem using KLD as the divergence, with no restrictions on the space of distributions $P_\Theta(x)$ and with the loss function $\ell_{HM}(\theta, x)$.*

*Proof.* The Generalised variational inference method seeks to solve the following optimization problem:

$$\underset{q(\theta) \in P_\theta}{\arg\min} \quad \mathbb{E}_{q(\theta)}[\sum_{i=1}^{n} \ell_{HM}(\theta, x_i)] + \mathrm{KLD}(q(\theta)||p(\theta)). \tag{4.18}$$

If $\theta \notin \mathcal{N}$, $\ell_{HM}(\theta, x_i) = \infty$ for some $x_i$ ($\ell_{HM}(\theta, x_i) = 0$ *otherwise*). Then we must have the posterior $q_{HM}(\theta) = 0$ for $\theta \notin \mathcal{N}$ leading to the optimization:

$$\underset{\mathrm{supp}(q) \subseteq \mathcal{N}}{\arg\min} \mathrm{KLD}(q(\theta)||p(\theta)). \tag{4.19}$$

Let $\mathcal{Q} = \mathrm{supp}(q)$ denote the support of $q$. We need to minimise :

$$\mathrm{KLD}(q(\theta)\|p(\theta)) = \int_{\mathcal{Q}} q(\theta) \log \frac{q(\theta)}{p(\theta)} d\theta,$$

subject to $\int q(\theta)d\theta = 1$.

We can do this via the calculus of variations. Introduce a Lagrange multiplier $\lambda$ and define the objective to maximise to be

$$\mathcal{L} = -\int q(\theta) \log \frac{q(\theta)}{p(\theta)} d\theta - \lambda(\int q(\theta)d\theta - 1).$$

Denote $\mathcal{L} = \int L(\theta, q)d\theta$, we solve the Euler-Lagrange equation:

$$\begin{aligned} 0 &= \frac{\partial L}{\partial q} \\ &= -1 - \log q(\theta) + \log p(\theta) - \lambda \end{aligned}$$

and so $q(\theta) = p(\theta)\exp^{-\lambda-1}$. Since $1 = \int_{\mathcal{Q}} q(\theta)d\theta = \exp^{-\lambda-1}\int_{\mathcal{Q}} p(\theta)d\theta$, we substitute to obtain the posterior

$$q(\theta) = \frac{p(\theta)}{\int_{\mathcal{Q}} p(\theta)d\theta}, \quad \theta \in \mathcal{Q}.$$

All that remains is to show we must take $\mathcal{Q} = \mathcal{N}$ (we have already shown $\mathcal{Q} \subseteq \mathcal{N}$).

$$\mathrm{KLD}(q(\theta)\|p(\theta)) = \int_{\mathcal{Q}} \frac{p(\theta)}{|\mathcal{Q}|_p} \log \frac{1}{|\mathcal{Q}|_p} d\theta = -\log |\mathcal{Q}|_p,$$

where $|\mathcal{Q}|_p = \int_{\mathcal{Q}} p(\theta)d\theta$. We want to maximise $|\mathcal{Q}|_p$ and so take $\mathcal{Q} = \mathcal{N}$, which proves the proposition. □

It is worth noting that when the prior is uniformly distributed, then the HM posterior will be uniform. Conversely, in the case of a non-uniform prior, the posterior distribution is essentially the prior truncated to the NROY set and rescaled. The unique trait of the HM loss comes from the two distinct values for the History Matching loss, i.e. 0 and $\infty$. Practically speaking,

the inefficacy of the history matching loss becomes evident as it tends to entirely discard parameters that fall beyond the implausible threshold from the truth. Essentially, it can be viewed as a more restrictive version of the ABC, resulting in a highly constrained posterior.

## 4.5 M-open robustness

### 4.5.1 Intuitions

The loss functions listed in Table 4.1 represent key strategies used to achieve robust inference in statistics and machine learning, though they are not an exhaustive compilation. For the annealed loss function in GBI $w\ell$, it is clear to see that $w < 1$ encourages larger variances for the posterior (Agostinelli and Greco [2013]). Alquier et al. ([Alquier et al., 2016, Alquier and Ridgway, 2017]) demonstrate that these posteriors excel in capturing the true underlying distribution, particularly in situations where the true posterior distribution is complex or high-dimensional. The $\beta$D and $\gamma$D divergences have demonstrated their robustness with outliers (Cichocki and Amari [2010], Greco et al. [2008]). The derived $\beta$ and $\gamma$ losses are more robust than the log score whenever $\beta > 1$ (or $\gamma > 1$). These particular losses are convenient since they recover the negative log-likelihood as $\beta \to 1$ (or $\gamma \to 1$). The ABC and HM losses are easier to interpret since they explicitly consider discrepancies between simulated and observed data. The tolerance rate essentially measures the permissible level of error, enabling the posterior to retain information within this threshold and discard everything beyond it, either through a smoothing kernel for the ABC loss $\ell_{ABC}(\theta, x)$ or a stricter rule such as the implausible measure for the History Matching loss $\ell_{HM}(\theta, x)$.

| Methods | $\ell(\theta, x)$ | $D$ | $\Pi$ |
|---|---|---|---|
| Standard Bayes | $-\log(p(x_i\|\theta))$ | KLD | $\mathcal{P}(\theta)$ |
| VI | $-\log(p(x_i\|\theta))$ | KLD | $\mathcal{Q}$ |
| Divergence-based Bayes | $\ell^\beta$ and $\ell^\gamma$ | KLD | $\mathcal{Q}$ |
| History Matching | $\ell_{HM}(\theta, x)$ | KLD | $\mathcal{Q}$ |
| ABC | $\ell^h_{ABC}(\theta, x)$ | KLD | $\mathcal{Q}$ |

Table 4.1: Generalised Bayesian settings considered (Equation 2.23) in Chapter 4.

### 4.5.2 Robustification Strategy

The Generalised Bayesian setting, denoted as $\mathcal{P}(\ell, \text{KLD}, \Pi)$, provides a convenient framework for straightforwardly comparing the robustness performance of various losses. Within this framework, the robustness of the inference strategy hinges solely on the choice of the loss function, as the divergence and the distribution family remain constant. We leverage the variational autoencoder discussed in Chapter 3 across diverse loss functions. This implementation entails modifying line 12 in Algorithm 1, replacing $\mathcal{L}_{recon}$ with the losses specified in Section 4.1. For each loss function, a non-exhaustive list of hyperparameters is selected, and the simulations are conducted multiple times for each generalised Bayesian setting. The hyperparameters for the $\ell^w$, $\ell^\gamma$, $\ell^\beta$ and $\ell^h_{ABC}$ losses are respectively $w$, $\gamma$, $\beta$ and $h$. In Section 4.7, we present a detailed examination of the performance of the proposed loss functions. This will be achieved by comparing the consistency and uncertainty obtained for each Generalised Bayesian posterior.

Moreover, we aim to draw a parallel with the most common scenario used to demonstrate the success of robust strategies, which is examining the impact of contaminated data. Contamination typically occurs when a small portion of the data is distributed according to another unknown distribution within the dGp. In Section 4.7.4, we randomly include outliers in the observations without introducing another distribution within the dGp.

While the model is not misspecified, a small portion of the data is contaminated, which, in our view, presents a more intriguing setting for studying contaminated data in dynamic modelling.

## 4.6 Misspecified ODE-based models

The goal is to showcase the effectiveness of the proposed loss functions through the exploration of misspecified ODE-based models. We utilise two models from Chapter 3, described in detail in Section 3.5, including the free fall model (Example 3.4.1) and the simple harmonic motion model (Example 3.4.3). As a reminder, the true dGp is defined for both models with a level of error denoted as $\delta$:

$$\frac{d^2x}{dt^2} = f_\theta(x,t) + \delta m(x)$$

where $f_\theta(x,t)$ represents the ODE model fitted to the data within the Variational Auto Encoding framework, as explained in Section 3.2.4 and $m(x)$ is the unknown misspecified dynamic independent of $\theta$. For both models, we generate data with $m(x) = -\frac{dx}{dt}$ and $\delta = \{0.0, 0.05, 0.1, 0.2\}$. If $\delta = 0$, the model exactly corresponds to the true dGp. When $\delta \neq 0$, the ODE model fitted against the data lacks the representation of the unknown dynamics $m(x)$. As the parameter $\delta$ increases, it signifies a higher level of error, indicating a greater degree of misspecification.

We will evaluate the performance of each GVI robust strategy by assessing $\theta^{\text{true}} - \theta^*_{GVI}$, where $\theta^{\text{true}}$ represents the parameter used to generate the observations, and $\theta^*_{GVI}$ is the estimator obtained using a robust loss $\ell$ for an error level $\delta$. A lower value of this difference indicates a more robust inference performance. Since we investigated various levels of errors

denoted as $\delta$, we can assess the robustness of each Bayesian setting by examining the effect of misspecification on the estimates for each error. We observe a connection with the global robustness (Sivaganesan [2000]) concept, which refers to the property of a method to maintain its performance or effectiveness even when subjected to variations or uncertainties. We can empirically evaluate whether each loss exhibits sensitivity to perturbations of the dynamical model.

## 4.7 Benchmark Results

In this section, all the comparison plot displays the standard deviation for the variational mean obtained, with the simulations repeated at least 5 times.

### 4.7.1 Ineffectiveness of Gibbs posteriors

As observed in Figure 4.1 and Figure 4.2, the log loss and the annealed loss achieve a close approximation to the true posterior when the model is correctly specified, as seen in Chapter 3. This alignment indicates the effectiveness of these loss functions in capturing the underlying data distribution when the model is accurately specified. As the level of misspecification $\delta$ grows, the Gibbs posterior (just as the logarithmic loss) exhibits a rapid decline in consistency, yielding inconsistent posterior estimates with relatively small variances. To exemplify using the Table 4.2, in the case where the model misspecification error is $\delta = 0.1$, the variational Gibbs posterior, labeled as $q^{GB}_{w=0.95}(\theta)$ the wrong parameter ($\bar{\mu} = 4.12$) from the truth ($\theta = 10$) with a high level of confidence ($\bar{\sigma} = 1.64$). Similar findings persist across all chosen hyperparameters for the annealing process highlighting its sensitivity to model misspecification. The variational variance obtained with the Gibbs posterior is stable (approximately $\sigma = 0.002$)

with increasing levels of misspecification, highlighting the lack of robustness for this loss function. Furthermore, there is no discernible advantage in altering the value of $w$ for the annealed posterior. No improvement is observed in the estimates for both the mean and variance, and there is no apparent reduction in sensitivity, even when down-weighting the likelihood significantly ($w = 0.2$).

### 4.7.2 Classical robust losses

In contrast, the $\gamma$ and $\beta$ losses produce slightly improved UQ around the estimates, as demonstrated in Figure 4.1 and Figure 4.2. Although exhibiting increasing bias similar to the logarithmic loss function, the $\ell^{(\gamma)}$ and $\ell^{(\beta)}$ losses provide improved uncertainty quantification when the model is misspecified. We observe this trend across all sets of hyperparameters $\gamma$ and $\beta$ used for both losses. These losses outperform the log loss across all levels of misspecification, as indicated by $\delta$, and consistently yield higher variance as shown in Table 4.2. Similarly, for the $\beta$ loss, we notice slightly more biased estimators with no discernible improvements even with a higher value for the hyperparameter $\beta$ in Figure 4.1. We faced convergence challenges with the $\ell_\beta$ loss, which was diverging towards infinity. This divergence is often caused by using a learning rate that is too high. The model's parameter updates become overly aggressive, leading to an uncontrollable increase in the loss function. We decided to lower the learning rate to 0.2 specifically for the $\ell^{(\beta)}$ loss. The increase in data size from 20 (Table 4.2) to 50 data points (Table 4.3) has contributed to a slight reduction in bias across all losses, with the robust losses, $\beta$, and $\gamma$, further emphasising their robust performances. The improved consistency observed with the $\gamma$ loss in Figure 4.1 across different levels of errors may be attributed to its robustness properties. The $\gamma$ loss is designed to down-weight outliers more aggres-

sively compared to other loss functions, thereby reducing their influence on the parameter estimates. This enhanced robustness helps to mitigate the impact of errors, resulting in more consistent estimators across varying error levels.

| | | $\delta = 0$ | | $\delta = 0.05$ | | $\delta = 0.1$ | | $\delta = 0.2$ | |
|---|---|---|---|---|---|---|---|---|---|
| | | $\bar{\mu}$ | $\bar{\sigma}^2$ | $\bar{\mu}$ | $\bar{\sigma}^2$ | $\bar{\mu}$ | $\bar{\sigma}^2$ | $\bar{\mu}$ | $\bar{\sigma}^2$ |
| Log Loss | $w = 1$ | 10.02 | 0.002 | 7.11 | 4.28 | 6.48 | 4.38 | 5.29 | 4.7 |
| $\ell^{(w)}$ | $w = 0.95$ | 10.04 | 0.01 | 6.2 | 1.43 | 4.12 | 1.64 | 5.25 | 4.25 |
| | $w = 0.5$ | 10.05 | 2.72 | 7.92 | 2.24 | 5.96 | 1.59 | 5.56 | 0.04 |
| | $w = 0.2$ | 10.06 | 2.71 | 6.33 | 2.6 | 5.64 | 2.5 | 5.25 | 4.56 |
| $\ell^{(\gamma)}$ | $\gamma = 1.01$ | 9.9 | 21.3 | 7.15 | 9.88 | 7.08 | 10.65 | 6.89 | 12.68 |
| | $\gamma = 1.05$ | 10.33 | 24.06 | 9.15 | 0.01 | 8.79 | 0.05 | 8.74 | 4.58 |
| | $\gamma = 1.1$ | 8.49 | 0.31 | 9.99 | 0.25 | 8.69 | 0.02 | 6.08 | 11.57 |
| $\ell^{(\beta)}$ | $\beta = 1.01$ | 8.73 | 0.21 | 8.3 | 3.67 | 8.18 | 63.32 | 6.86 | 5.39 |
| | $\beta = 1.05$ | 6.77 | 1.94 | 7.6 | 0.64 | 6.81 | 3.33 | 6.38 | 72.75 |
| | $\beta = 1.11$ | 8.03 | 5.11 | 8.75 | 8.08 | 7.62 | 174.19 | 7.16 | 0.21 |

Table 4.2: Variational posteriors mean and variance averages obtained for the misspecified free fall model for a set of hyperparameters $w, \gamma, \beta$ across three levels of misspecification $\delta = \{0.05, 0.1, 0.2\}$ and with the correct model ($\delta = 0$). The simulations are run a minimum of 5 times for each scenario against $n = 20$ observations.

| | | $\delta = 0$ | | $\delta = 0.05$ | | $\delta = 0.1$ | | $\delta = 0.2$ | |
|---|---|---|---|---|---|---|---|---|---|
| | | $\bar{\mu}$ | $\bar{\sigma}^2$ | $\bar{\mu}$ | $\bar{\sigma}^2$ | $\bar{\mu}$ | $\bar{\sigma}^2$ | $\bar{\mu}$ | $\bar{\sigma}^2$ |
| Log Loss | $w = 1$ | 10.01 | 0.002 | 6.69 | 1.84 | 6.17 | 1.84 | 5.14 | 3.66 |
| $\ell^{(w)}$ | $w = 0.95$ | 10.51 | 0.01 | 5.2 | 0.21 | 5.24 | 2.23 | 4.79 | 2.5 |
| | $w = 0.5$ | 9.41 | 3.14 | 6.88 | 3.98 | 5.95 | 3.13 | 5.64 | 0.18 |
| | $w = 0.2$ | 10.15 | 2.72 | 6.1 | 0.03 | 6.19 | 3.7 | 6.77 | 4.13 |
| $\ell^{(\gamma)}$ | $\gamma = 1.01$ | 10.56 | 61.3 | 12.35 | 20.19 | 9.89 | 0.03 | 10.27 | 0.12 |
| | $\gamma = 1.05$ | 7.87 | 6.5 | 10.53 | 0.25 | 8.67 | 37.61 | 11.64 | 4.18 |
| | $\gamma = 1.1$ | 10.13 | 2.2 | 8.44 | 0.06 | 9.31 | 5.92 | 8.53 | 0.02 |
| $\ell^{(\beta)}$ | $\beta = 1.01$ | 10.34 | 0.15 | 8.92 | 2.65 | 13.52 | 273.97 | 10.58 | 0.54 |
| | $\beta = 1.05$ | 8.32 | 19.28 | 9.1 | 2.07 | 9.13 | 0.13 | 8.59 | 40.28 |
| | $\beta = 1.11$ | 11.13 | 23.11 | 9.72 | 4.1 | 8.14 | 0.95 | 9.01 | 35.21 |

Table 4.3: Variational posteriors mean and variance averages obtained for the misspecified free fall model for a set of hyperparameters $w, \gamma, \beta$ across three levels of misspecification $\delta = \{0.05, 0.1, 0.2\}$ and with the correct model ($\delta = 0$). The simulations are run a minimum of 5 times for each scenario against $n = 50$ observations.

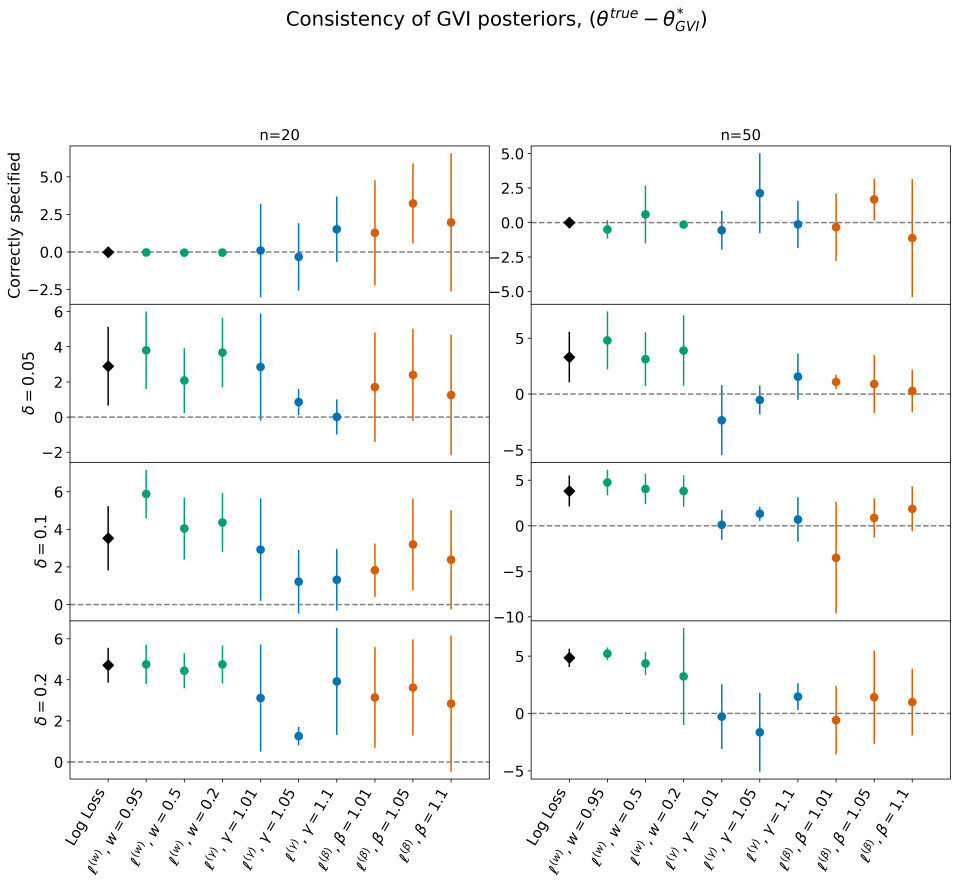



Figure 4.1: Comparing standard VI against GVI with the $\ell^w$ $\ell^\beta$ and $\ell^\gamma$ losses for the free fall model with 20 and 50 data points. The $y$-axis quantifies the difference between the posterior belief and the truth $(\theta^* - \theta_{truth})$. Dots and whiskers represent posterior means and their respective standard deviations for each posterior with different values for the hyperparameters $w$, $\beta$, and $\gamma$ across multiple levels of misspecification $\delta$, including the correct model.

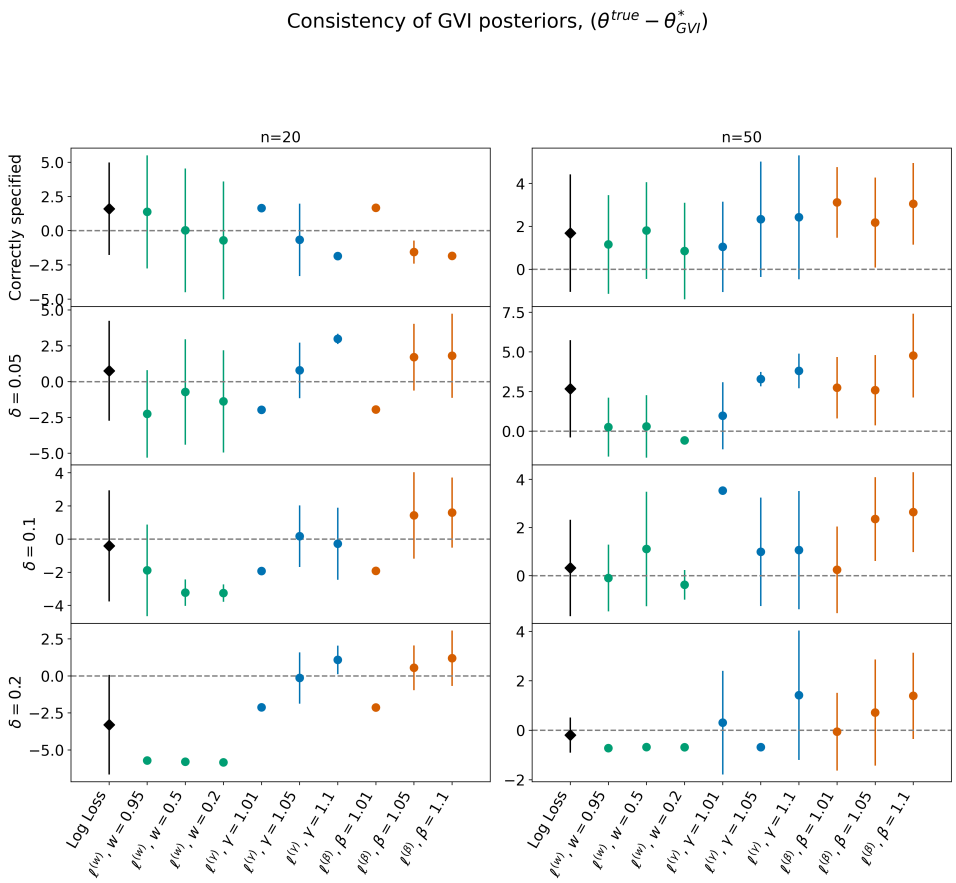



Figure 4.2: Comparing standard VI against GVI with the $\ell^w$ $\ell^\beta$ and $\ell^\gamma$ losses for the pendulum model with 20 and 50 data points. The $y$-axis quantifies the difference between the posterior belief and the truth $(\theta^* - \theta_{truth})$. Dots and whiskers represent posterior means and their respective standard deviations for each posterior with different values for the hyperparameters $w$, $\beta$, and $\gamma$ across multiple levels of misspecification $\delta$, including the correct model.

### 4.7.3 Approximate Bayesian Computation derived loss

With the observations $x_{1:n}$ and a tolerance rate $h \in \mathbb{R}^+$, we take a simple uniform kernel for the ABC loss (Equation 1) and we follow the steps:

- Generate $\theta$ from the prior $p(\theta)$ (Variational Encoder)
- Generate $y \sim p(y \mid \theta)$ from the likelihood (Variational Decoder).
- Use the ABC loss in the variational optimization so that :

$$\ell^h_{ABC}(\theta, x) = -\log(I(|y - x| \leq h)p(y|\theta))$$

  where $I$ is the indicator function, $\sigma^2$ is the known data noise and $h$ the tolerance rate.

The ABC loss function $\ell^h_{ABC}(\theta, x)$ facilitates a meticulous selection of the parameter $\theta$, ensuring that the predictions obtained with the ODE solver closely match the true observations within a tolerance $h$. We can arbitrarily choose a smaller value for $h$ to impose high restrictions on the posterior estimate for $\theta$. We choose a minimum value of 5 for $h$, as below that threshold, the number of data points is reduced to less than 10% of the total initial data size. In Figure 4.3, the resulting posteriors provide a more accurate estimate compared to the $\ell^\beta$, and $\ell^\gamma$ losses when the model is correctly specified for $n = 50$. When the model is slightly misspecified ($\delta = 0.05$), the estimate remains fairly consistent across different values of the hyperparameter $h$. When higher structural error is included ($\delta > 0.05$), we observe comparable bias to classical robust losses, but with reduced variance. This is related to the form of the chosen loss function, which preserves the logarithmic loss within a tolerance $h$, resulting in overly confident estimates.

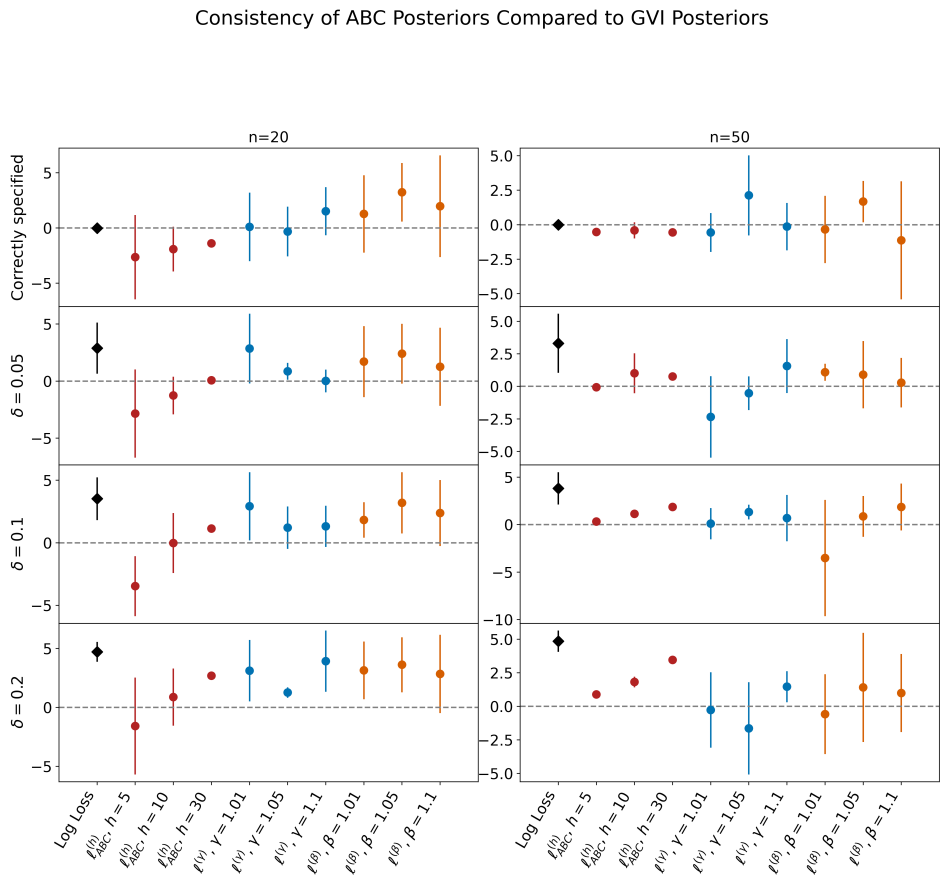



Figure 4.3: Comparing the ABC loss $\ell^h_{ABC}(\theta, x)$ against the $\ell^\beta$ and $\ell^\gamma$ robust losses for the free fall model with 20 and 50 data points. The $y$-axis quantifies the difference between the posterior belief and the truth $(\theta^* - \theta_{truth})$. Dots and whiskers represent posterior means and their respective standard deviations for each posterior with different values for the hyperparameters $h$, $\beta$, and $\gamma$ across multiple levels of misspecification $\delta$, including the correct model.

### 4.7.4 Dynamical Model Contamination

As explored in Chapter 2, robust inference aims to produce accurate estimators in the presence of data contamination. We compare the best robust losses against the logarithm loss for the Free Fall model where we contaminate the dataset with $\epsilon\%$ outliers. In theory, the robust losses examined here are more resilient to these types of misspecification.

This perspective differs from the classical $\epsilon$-contamination problem discussed in Chapter 2, where another unknown distribution generates a small portion of the data. However, when dealing with spatial dynamics, the nature of corruption in dynamical models is often more nuanced. In such cases, the deviations from the expected trajectories are not necessarily caused by a distinct, contaminated subset of data. Instead, these deviations arise due to various factors such as measurement errors, modelling inaccuracies, or environmental influences. Therefore, the classical $\epsilon$-contamination perspective may not be the most suitable for understanding and mitigating corruption in dynamical models, where deviations from expected trajectories are typically more complex and subtle.

We consider data simulated via the correctly specified dynamical model Example 3.4.1) and corrupt $\epsilon\%$ of the data where we replace the observation $x_i$ by $(x_i + \sin(t_i) \times 30)$ at time $t_i$. This enables us to perturb the corrupted observations slightly within a small range of the true observations. Figure 4.4 demonstrates that the log-loss is not very sensitive to a small outlier contamination of 3%, but it tends to overestimate the parameters when the corruption proportion of data is higher. Conversely, the $\beta$ and $\gamma$ robust losses exhibit higher variance in both scenarios, with reduced bias observed when 15% of the data are contaminated. The ABC loss is particularly useful in contamination scenarios, as it can generate consistent estimators across any level of contamination. This is because the level of

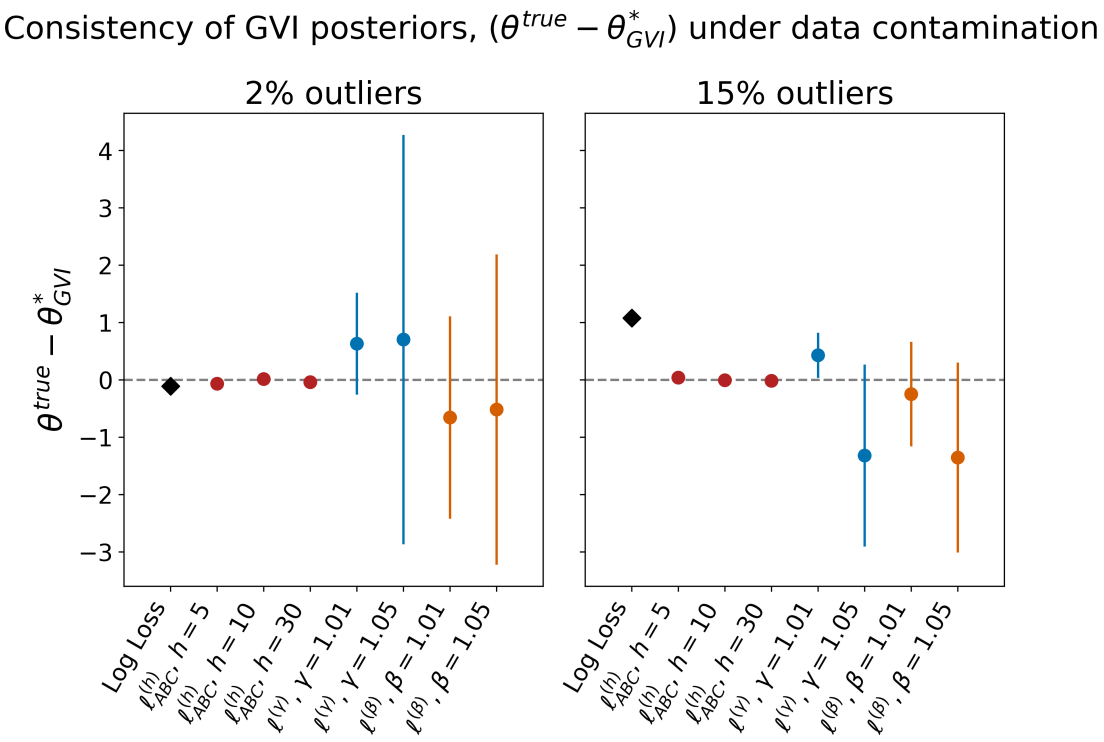



Figure 4.4: Comparing standard VI against GVI with the $\ell^h_{ABC}(\theta, x)$, $\ell^\beta$ and $\ell^\gamma$ losses for the free fall model with $n = 100$ data points with 3% outliers and 15% outliers.

tolerance, carefully chosen as $h \leq 30$ (which corresponds to the contamination error considered here), allows the ABC loss to discard values that are too far from the observations. The ABC loss offers superior robustness compared to other losses, as demonstrated in Figure 4.4.

## 4.8 Limitations

Many problems in statistics and optimization require robustness. This idea is common in parameter estimation and learning tasks, where a robust loss (say, absolute error) may be preferred over a non-robust loss (say, squared error) due to its reduced sensitivity to large errors as explained in Section 2.4 in Chapter 2. In Bayesian statistics, the standard likelihood-based formulation, relying on negative log-likelihood loss, can yield misleading

estimates with structural model misspecification as observed in Chapter 3. In this chapter, we have chosen well-established losses from recent literature that claim to be robust against model misspecification ([Knoblauch et al., 2018, Medina et al., 2022]). We observed slightly better robustness against misspecification in both bias and uncertainty quantification with the $\beta$ and $\gamma$ losses. However, the results are not as convincing compared to the promising benefits reported in the literature (Cichocki and Amari [2010], Jewson et al. [2018]). We note some improvement with increasing hyperparameters, especially with the $\gamma$ loss, which raises questions about the impact of hyperparameter tuning on the performance of these robust losses. Investigating the optimal choice of hyperparameters and understanding their influence on the robustness and performance of the Bayesian setting could be a valuable avenue for future research (Jewson et al. [2023]). This exploration may provide insights into achieving a balance between downweighting the likelihood and obtaining more accurate and robust posterior estimates. Despite yielding interesting results, the classical robust losses demonstrated poor performance when confronted with the relatively simple ODE models selected in this study challenging the universal effectiveness of these robust losses. To our knowledge, this study represents one of the initial applications of Generalised Variational Inference for investigating misspecified dynamical examples. The various concerns related to the efficiency and robustness of Bayesian inference in the context of model misspecification have prompted useful alternatives relying on ABC ([Frazier, 2020, Schmon et al., 2020]) ). We opted to incorporate an ABC loss and showcase that this methodology can produce robust estimates. Specifically, when the model poorly describes $y$ for a certain value of $\theta$, incorporating a tolerance rate can fortify the inference process. We establish a natural connection between ABC and generalised Bayesian approaches, reinterpreting it as a flexible robustification strategy.

These findings also underscore the challenges of selecting specific loss functions in cases where misspecification involves structural errors rather than outliers.

## 4.9 Further work

The Generalised Bayesian Inference is considered due to its distinct loss function-based formulation compared to the likelihood-based approach. Another likelihood-free approach involves comparing empirical distributions without explicitly estimating the underlying probability density functions. For example, the Maximum Mean Discrepancy (MMD) distance (Gretton et al. [2006]) on the space of probability measures has found numerous applications in machine learning and nonparametric testing. This distance is based on the notion of embedding probabilities in a reproducing kernel Hilbert space and can be used for inference (Briol et al. [2019]). Alternatively, one can consider the Stein Discrepancy (Stein [1972]); for instance, Barp et al. [2019] introduces a Stein score estimator, utilising the Hyvärinen scoring rule (Hyvarinen [2005]), which has exhibited robustness when exposed to corrupted data. These techniques refrain from employing a predefined model structure and instead rely solely on samples from the dGp. Although they can achieve better proximity to the true distribution, they do not offer a mechanism for integrating mechanistic modelling with interpretable parameters. Chapter 5 presents a novel methodology for misspecified mechanistic models which involves adopting a variational approach with an augmented Gaussian process model.

# Chapter 5

# Using Gaussian Processes to mitigate mechanistic model misspecification.

**Summary:** Many scientific disciplines use models formulated as ordinary differential equations, derived from fundamental principles and mechanistic insights. Typically, an ODE model takes the form of $\frac{dx}{dt} = f_\theta(x, t)$ where $x$ represents time-varying variables, and $\theta$ denotes static parameters of the ODE model $f$. Inaccurately representing the dynamics with the model poses significant challenges for both inference and uncertainty quantification for $\theta$. This study introduces a novel approach that integrates Gaussian processes into the ODE model, inferred via a variational inference framework. We underscore the importance of incorporating model discrepancy to capture mechanistic dynamics and emphasise the effectiveness of our proposed method for robust inference and prediction. The study aims to contribute to a better understanding of how model discrepancy should be appropriately modeled in the context of mechanistic misspecified models.

## 5.1 Introduction

Dynamic processes are frequently modeled using ordinary differential equation models (ODEs). However, inferring the parameters of the ODE system becomes challenging when the model fails to capture all relevant mechanisms, leading to what is known as model misspecification. The extensive exploration conducted in Chapters 3 and 4 illustrated that even minor perturbations in the model can lead to misleading estimations and, consequently unreliable predictions when relying on a classical Bayesian design ([Grünwald and van Ommen, 2014, Broderick et al., 2023]). Model discrepancy, which refers to the difference between the model and the true data-generating process, introduces bias in inference and compromises the accuracy of predictions. This suboptimal performance becomes significant when higher uncertainty in mechanistic models is considered, leading toward a non-robust framework.

Our main findings from the previous chapters, where we learn parameters from observations of a misspecified mechanistic system, are summarised as follows:

- if model discrepancy is ignored, both predictions and inferences about parameters are biased, and this bias persists with increasing numbers of observations (*Bayesian inconsistency*);
- if model discrepancy is considered using a generalised Bayesian posterior, uncertainty quantification is improved but inferences about parameters will still typically be biased;
- to achieve better inference, a more careful modelling approach that explicitly considers model discrepancy is needed.

Model discrepancy was formally introduced as a source of uncertainty in simulator predictions by Kennedy and O'Hagan [2001], who referred to

it as model inadequacy. They addressed the challenge of incorporating uncertainty in simulator predictions when learning about uncertain input parameters from observations of the real physical system, a process known as calibration ([Goldstein and Rougier, 2009, Higdon et al., 2004]). Their work demonstrated how we can incorporate model discrepancy into calibration and subsequent predictions of the physical system. This modelling framework often referred to as the "Kennedy-and-O'Hagan approach" has been widely adopted in fields such as health sciences (Oakley and Youngman [2017]), experimental physics ([Wilkinson et al., 2011] ) or climate modelling (Murphy et al. [2007]).

The typical form for model discrepancy can be described using the terminology and notation introduced by Kennedy and O'Hagan [2001]. Let $z_{1:n}$ denote $n$ observations of the physical system, with the $i$-th observation $z_i$ associated with control inputs $x_i$. Each observation $z_i$ is modeled as $z_i = \zeta(x_i) + \epsilon_i$, where $\zeta(x_i)$ is the true value of the physical system at control variable value $x_i$, and $\epsilon_i$ represents independent observation errors. Model discrepancy is introduced with $\zeta(x) = \eta(x, \theta) + \rho(x)$, where $\theta$ is the true unknown parameter. We can employ a Bayesian approach, where prior distributions are assigned to $\theta$ and the model discrepancy function $\rho(\cdot)$, and these are updated to posterior distributions conditioned on the observations. The model discrepancy is typically inferred with a Gaussian Process (GP) model learned jointly with the model parameters $\theta$ where we map the model outputs $y$ and inputs $x$ to the observational data $z$: $\mathcal{GP} : \{y, x\} \to z$. GP regression is chosen for modelling model discrepancy due to its flexibility, nonparametric nature, and Bayesian formulation, allowing the estimation of uncertainty [Kennedy and O'Hagan, 2001, Conti et al., 2009, Pope et al., 2021, Gahungu et al., 2022]). Especially in the case of complex dynamical systems where misspecification is suspected, employing a flexible framework such as GP can handle noisy and incom-

plete data and account for uncertainty in the model predictions ([Lei et al., 2020, Zhou et al., 2022, Coveney et al., 2022]).

We posit the introduction of a novel framework in which a discrepancy model is seamlessly incorporated alongside the mechanistic differential equation model. When we incorporate additional uncertainty in the model structure and learn about what is still unknown from the model, the underlying mechanistic model becomes :

$$\frac{dx}{dt} = f_\theta(x, t) + \rho(\cdot) \tag{5.1}$$

In this context, $\rho(\cdot)$ is considered the discrepancy between the reality and the model governed by ordinary differential equations at the best $\theta$, and it remains independent of the model parameters $\theta$. We assume that $f_\theta(x, t)$ is the known expert model for the dynamical process with $\theta$ meaningful physical parameters.

**Chapter Contributions** In this work, we propose a novel approach for uncertainty quantification in misspecified dynamical systems :

- We present an innovative approach to differentiate the GP through the ODE solver, departing from the classical trajectory outputs derived approach.
- We demonstrate the efficient integration of mechanistic knowledge and Gaussian Processes (GP) into a Variational Autoencoder (VAE) framework.
- We illustrate the benefits derived from integrating model discrepancy in both inference and prediction when the mechanistic model is misspecified.

## 5.2 Quantifying Uncertainty in Differential Equation Models

In this study, we expand upon the Kennedy and O'Hagan approach to estimate the parameter $\theta \in \mathbb{R}^d$ in mechanistic systems governed by ordinary differential equations denoted as $f_\theta(x,t)$. We introduce two discrepancy functions $\rho_1(x,t,z)$ and $\rho_2(z,t,x)$ involving two current time-dependent states $x$ and $z$ so that the general discrepancy takes the form:

$$\frac{dw}{dt} = \begin{bmatrix} \frac{dx}{dt} \\ \frac{dz}{dt} \end{bmatrix} = \begin{bmatrix} f_\theta(x,t) + \rho_1(x,t,z) \\ \rho_2(z,t,x) \end{bmatrix}. \tag{5.2}$$

The typical approach involves adding a discrepancy term $\rho(\cdot)$ directly to the observations. We seek to enhance the methodology by integrating the discrepancy term $\rho_1(x,t,z)$ within the ordinary differential equation solver itself. By directly accounting for discrepancies within the ODE solver, we expect to achieve more accurate and reliable parameter estimation and predictions in mechanistic systems.

The second discrepancy function $\rho_2(z,t,x)$ becomes particularly relevant when it is suspected that an equation is omitted from the ODE system (Misspecification of Type 3). In Chapter 2, the ion channel modelling example 3 illustrates this situation, where the blue model does not include the state $\frac{d[C_2]}{dt}$ present in the red model that generates the data. To model the discrepancy, we utilise a Gaussian process, denoted as $\rho(x) \sim \mathcal{GP}(m(x), k(x,x'))$, where $m(x)$ signifies the mean function, and $k(x,x')$ is the covariance function or kernel. The VAE architecture, as discussed in Chapter 3, offers a suitable framework for incorporating the differential equation in Equation 5.2. To overcome the need for gradient differentiation during backpropagation through the ODE solver (further details in Chapter 3, Section 3.3),

we propose employing Random Fourier Features (RFF) as an approximation method for Gaussian Processes (GPs) (Hensman et al. [2018]).

## 5.3 Background

As proposed by Rahimi and Recht [2007], we use random Fourier features (RFFs) to approximate the kernel function of a GP. This method leverages Bochner's theorem (Rudin [2011]), expressing stationary kernels $k(x, y) := k(x - y)$ as the Fourier transform of a positive measure $p$.

$$k(x - y) = \int \exp^{iw^T(x-y)} dp(w) = E_w[\zeta_w(x)\zeta_w(y)^*], \tag{5.3}$$

where $\zeta_w(x) = \exp^{iw^T x}$, and the superscript $*$ denote the complex conjugate. Importantly, recall that the complex conjugate of $\exp^{ix}$ is $\exp^{-ix}$. $\zeta_w(x)\zeta_w(y)^*$ is an unbiased estimate of the kernel $k(x, y)$ when $w$ is drawn from the distribution $p$. Since the probability distribution $p(w)$ and the kernel $k$ are real, the integral converges when the complex exponential is replaced with cosines:

$$k(x, y) = \mathbb{E}_w[\Phi_m(x)\Phi_m(y)] = \sum_{m=1}^{\infty} \Phi_m(x)\Phi_m(y), \tag{5.4}$$

with the basis vector $\Phi_m(x) = \sqrt{2}\cos(w_m^T x + b_m)$, $w_m$ is drawn from $p(w)$ and $b_m$ is drawn uniformly from the uniform distribution $U[0, 2\pi]$. This expression allows us to interpret the covariance function as an expectation that can be estimated using Monte Carlo. It follows that, with $D$ samples, we can estimate the covariance function as:

$$k(x, y) \approx \frac{1}{D}\sum_{m=1}^{D} \Phi_m(x)\Phi_m(y) = \frac{1}{D}\sum_{m=1}^{D} \cos(w_m^T(x - y)). \tag{5.5}$$

*Proof.* $\mathrm{E}_{w,b}[2\cos(w^T x + b)\cos(w^T y + b)]$
$= \mathbb{E}_{w,b}\left[\cos(w^T(x-y)) + \cos(w^T(x+y) + 2b)\right]$
$= \mathbb{E}_w\left[\cos(w^T(x-y))\right] + \mathbb{E}_w\mathbb{E}_b\left[\cos(w^T(x+y) + 2b)\right]$
$= k(x,y) + 0.$ □

We specifically select the radial basis function (RBF) covariance defined as:

$$k_{RBF}(x,y) = \exp\Big(-\frac{\|x-y\|^2}{2\lambda}\Big),$$

parameterised by a length scale parameter $\lambda > 0$ the length-scale. In the rest of this chapter, we choose $\lambda = 1$ and $p(w) = \mathcal{N}(0, I)$. This choice provides an initial perspective for the experiments, though we acknowledge the importance of exploring alternative kernels in future research. The RBF kernel can thus be estimated using the Monte Carlo approximation provided in Equation 5.5. To achieve a comprehensive representation of the spectrum, the RFF methodology typically requires a large number of spectral sample points. Figure 5.1 illustrates the RBF covariance matrix approximation using RFF. As $D$ increases, meaning the Monte Carlo approximation employs more samples to estimate the basis function $\Phi(\cdot)$, we achieve a better approximation for the covariance matrix.

## 5.4 Methodology

### 5.4.1 Augmented Dynamical Model

In the upcoming section, we delve into the task outlined in Section 5.2 with a more specific form for the dynamic model, which will be later illustrated in the results section. We consider the dynamics of a system governed by the ordinary differential equation $f_\theta(\cdot)$, with an additional Gaussian

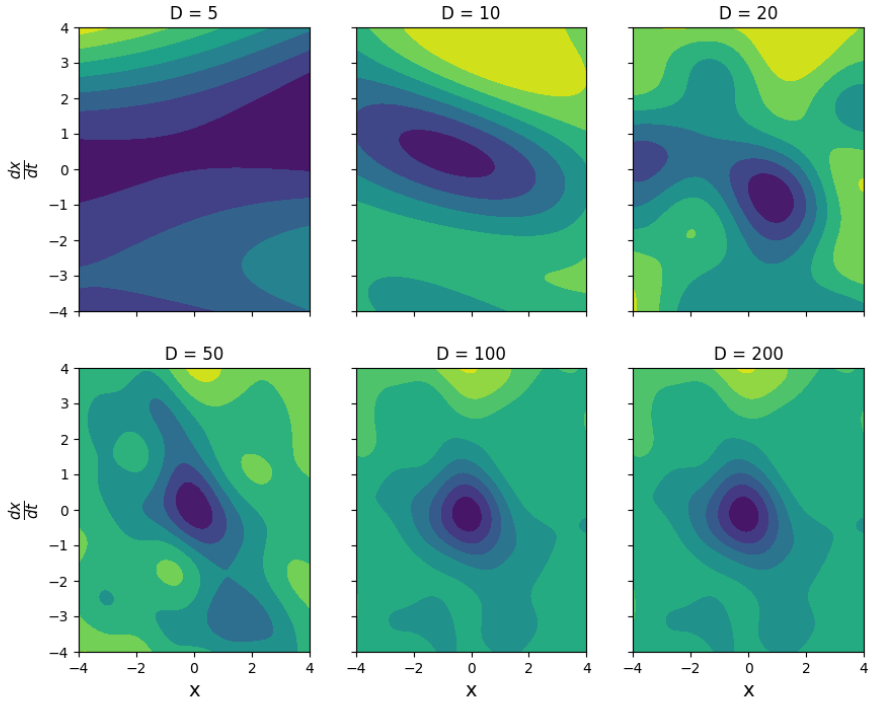



Figure 5.1: Random Fourier Features kernel approximation for the RBP kernel with an increasing number of Monte Carlo samples given by $D$. We conclude that increasing $D$ beyond 200 offers no clear benefit based on initial experiments, though this is not definitive.

Process (GP) term denoted as $\rho(\cdot)$ independent of $\theta$:

$$\frac{d^2x}{dt^2} = f_\theta(x,t) + \rho(x).$$

We choose to model the discrepancy as a zero-mean Gaussian Process (GP) with an RBF kernel with length scale $\lambda = 1$, which will be approximated using Random Fourier Features (RFFs):

$$\rho(x) \sim \mathcal{GP}(0, k_{RBF}(x, x'))$$

The discrepancy model is intentionally made independent of $\theta$ and encompasses all the unknown aspects of the dynamic system, related to the mechanistic model $f$.

Furthermore, we choose to include the derivative states $\frac{dx}{dt}$ within the discrepancy function, in addition to the states $x$. We believe that this inclusion enables a more comprehensive representation of system dynamics. By considering both states and their derivatives, the model can better capture temporal changes and system behaviour over time. The model discrepancy is therefore characterised by a linear dependence on unknown parameters $\alpha_{1:m}$ with:

$$\rho(z) = \sum_{m=1}^{D} \alpha_m \Phi_m(z), \quad z = \left(x, \frac{dx}{dt}\right). \tag{5.6}$$

Following the previous section, substituting $\Phi_m$ in the Monte Carlo approximation enables the representation of $\rho(\cdot)$ without explicitly calculating the true Gaussian process as :

$$\rho(z) = \sum_{m=1}^{D} \alpha_m \cos(a_m^T z + b_m). \tag{5.7}$$

$a_m^T \in \mathbb{R}^{D\times 2}$ is drawn from $\mathcal{N}(0, I)$ and $b_m \in \mathbb{R}^D$ is drawn uniformly from the uniform distribution $U[0, 2\pi]$. $D$ represents the number of samples in

the Monte Carlo estimator.

**Example 8.** *Denote the true unknown dynamical model given by :*

$$\frac{d^2x}{d^2t} + \delta\frac{dx}{dt} + \theta\sin(x) = 0, \tag{5.8}$$

*where $\delta$ is a constant parameter in $\mathbb{R}$.*

*Suppose that the model used to describe the data is characterised by the (simpler) incorrect dynamics:*

$$\frac{d^2x}{d^2t} + \theta\sin(x) = 0, \tag{5.9}$$

*We define* $z(t) = \begin{bmatrix} x(t) \\ \dfrac{dx}{dt} \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$ *so that the true system is*

$$\frac{dz}{dt} = \begin{bmatrix} \dfrac{dx}{dt} \\ \dfrac{d^2x}{d^2t} \end{bmatrix} = \begin{bmatrix} \dfrac{dz_1}{dt} \\ \dfrac{dz_2}{dt} \end{bmatrix} = \begin{bmatrix} z_2 \\ -\theta sin(z_1) + \rho(z) \end{bmatrix}. \tag{5.10}$$

*Here $\rho(z) = -\delta z_2 = -\delta\frac{dx}{dt}$ represents the unknown discrepancy that we want to infer using GP.*

### 5.4.2 Discrepancy Modelling with RFF-VAE

In this work, we combine the variational autoencoder approach with Fourier features; we refer to our method in Algorithm 2 as Random Fourier Features Variational autoencoder (RFF-VAE). Discrepancy Modelling with RFF-VAE involves a variational Bayesian posterior distribution $q(\theta, \alpha_{1:D} \mid x)$ for the latent parameter $\theta$ within the dynamical model $f_\theta(x, t)$, where $\alpha_{1:D}$ are the parameters needed for the RFF approximation. We choose a mean field family $\mathcal{Q}$ for the variational posterior, i.e. we use $q$ of the form :

$$q(\theta, \alpha) = \mathcal{N}\left( \begin{bmatrix} \mu_\theta \\ \mu_{\alpha_{1:D}} \end{bmatrix}, \begin{bmatrix} \sigma_\theta^2 & 0 \\ 0 & diag(\sigma_{\alpha_{1:D}}^2)) \end{bmatrix} \right), \tag{5.11}$$

where $\mu_\phi = (\mu_\theta,\ \mu_{\alpha_{1:D}})$ are the variational posterior mean and $\sigma_\phi^2 = (\sigma_\theta^2,\ \sigma_{\alpha_{1:D}}^2)$ the variational posterior variance.

Variational Bayes involves maximising the Evidence Lower Bound (ELBO) $\mathcal{LB}(\phi)$, which is:

$$\mathcal{LB}(\phi) = \mathbb{E}_{q(\phi)}[\log p(y \mid x, \sigma^2)] - \mathrm{KLD}(q_\phi || p(\theta, \alpha)), \tag{5.12}$$

where $\phi = (\mu_\phi, \sigma_\phi^2)$ are variational parameters and $p(\theta, \alpha)$ a multivariate Gaussian prior distribution.

The first term in the ELBO is obtained by solving the augmented dynamical system $f_\theta(x, t) + \rho(z)$ with the reparametrization trick:

$$\begin{aligned} \epsilon &\sim \mathcal{N}(0, \boldsymbol{I}) \\ \theta &\leftarrow \mu_\theta + \sigma_\theta \circ \epsilon \\ \alpha_{1:D} &\leftarrow \mu_{\alpha_{1:D}} + \sigma_{\alpha_{1:D}} \circ \epsilon \\ x &\leftarrow \mathrm{ODESolve}\Big( f_\theta + \rho(z), x(t_0) \Big) \end{aligned} \tag{5.13}$$

where $\rho(z)$ is given in Equation 5.7.

Solving the dynamical system with the parameter results in the vector $x$ assumed normal with mean the observations $y$ and a variance noise $\sigma^2$ (assumed known).

The second term requires the computation of the KLD between two multivariate diagonal Gaussian distributions. The analytic expression is given in the Appendix (Proposition 3). In our optimization process, we employ `torch.distributions.kl.kl_divergence` from Pytorch for this computation.

**Algorithm 2** Discrepancy Modelling with RFF-VAE

**Input** : model $f_\theta(x,t)$, dataset $y_{1:n}$, prior $p(\theta,q)$, sampling size J, learning rate $\lambda$, size of RFF approximation $D$, vector $b_m \sim U[0,2\pi]$, vector $a_m^{1:2} \sim \mathcal{N}(0,1)$
**Initialize** : Variational parameter $\phi$ randomly.
**repeat**
  **Encoder**
  Draw samples from $\theta^j, \alpha_{1:D}^j \sim \mathcal{N}(\mu_\phi, \sigma_\phi^2 I)$ for $j = 1, \cdots, J$
  **Decoder**
  - Compute Basis vectors $\Phi(z) = \cos(a_m^{1:2} z + b_m)$ with $z = (x, \frac{dx}{dt})^\top$
  - Forward simulate $x^j \sim f_{\theta^j}(x,t) + \Phi(z)^T \alpha^j$ for $j = 1, \cdots, J$
  **Compute ELBO**
  $\widehat{\mathcal{LB}(\phi)} = \frac{1}{J}\sum_{j=1}^{J} \log p(y \mid x_j) - \mathrm{KLD}(q_\phi(\theta,q) \| p(\theta,q))$
  **Update parameter** $\phi = \phi + \lambda \nabla_\phi \widehat{\mathcal{LB}(\phi)}, \quad t = t+1$
**until** change of $\phi$ is small enough.
**return** $q_\phi(\theta)$

### 5.4.3 Implementation

The RFF-VAE framework given in Algorithm 2 is implemented in PyTorch where the encoder and differential function parameters are jointly optimized with the Adam optimizer (learning rate $\lambda = 0.4$). The sampling size for the Monte Carlo of the gradient ELBO is $J = 15$. The backpropagation through the Ordinary Differential Equation (ODE) solver, which incorporates both the dynamical system and the model discrepancy, is done with `torchdiffeq` (Chen et al. [2018]) for gradient computation. The ODE solver used is the adaptative `dopri5`. To use variational Fourier Features, we must select the vector components $[a, b]$ drawn from a Gaussian distribution $\mathcal{N}(0,1)$ and a uniform distribution $U[0,2\pi]$, respectively. The size of the RFF expansion approximation, denoted as $D$, will be varied in our approach, spanning from 10 to 200. While the RFF expansion requires more parameters for inference, the computational complexity of the reverse approach is primarily influenced by the number of steps in the ODE solver leading to similar computational time than in Chapter 3. We compare Variational Bayes with Hamiltonian Monte Carlo (HMC), a

Markov chain Monte Carlo (MCMC) method that uses the derivatives of the density function being sampled to facilitate efficient transitions across the posterior distribution (for further insights, Betancourt [2017]). We use the No-U-Turn sampler (NUTS) for HMC in Stan (Carpenter et al. [2017]).

## 5.5 Experiments

To demonstrate the effectiveness of the RFF-VAE, we evaluate our approach with several dynamical ODE models with noisy data coming from a true data-generating process. The mechanistic model is intentionally misspecified with various perturbations, evaluating parameter inference (bias and uncertainty quantification) and predictive performances. In all the presented results, we repeat the variational optimization several times to obtain an average estimate of the variational posterior.

You can find the comprehensive introduction to the dynamical models employed in the following section in Chapter 3, particularly in Section 3.4, along with the variational framework architecture. Table 5.2 presents the various models alongside the respective data-generating processes utilised to generate data for fitting. It is worth noting that the results from Chapter 3 were obtained using data generated from dGp A, B, and C, fitted respectively against models A1, B1, and C1.

| Dynamical Model | Data Generating Process | Misspecifed Model | Misspecifed Model + RFF-VAE | Well specified model | Well specified Model + RFF-VAE |
|---|---|---|---|---|---|
| A | dGp A $\frac{d^2x}{dt^2} = -\theta - \delta\frac{dx}{dt}$ | Model A1 $\frac{d^2x}{dt^2} = -\theta$ | Model A2 $\frac{d^2x}{dt^2} = -\theta + \rho(z)$ | Model A3 $\frac{d^2x}{dt^2} = -\theta - \delta\frac{dx}{dt}$ | Model A4 $\frac{d^2x}{dt^2} = -\theta - \delta\frac{dx}{dt} + \rho(z)$ |
| B | dGp B $\frac{d^2x}{dt^2} = -\theta\sin(x) - \delta\frac{dx}{dt}$ | Model B1 $\frac{d^2x}{dt^2} = -\theta\sin(x)$ | Model B2 $\frac{d^2x}{dt^2} = -\theta\sin(x) + \rho(z)$ | Model B3 $\frac{d^2x}{dt^2} = -\theta\sin(x) - \delta\frac{dx}{dt}$ | Model B4 $\frac{d^2x}{dt^2} = -\theta\sin(x) - \delta\frac{dx}{dt} + \rho(z)$ |
| C | dGp C $\frac{d^2x}{dt^2} = -\theta\sin(x) - \delta\frac{dx}{dt}$ | Model C1 $\frac{d^2x}{dt^2} = -\theta x$ | Model C2 $\frac{d^2x}{dt^2} = -\theta x + \rho(z)$ | Model C3 $\frac{d^2x}{dt^2} = -\theta x - \delta\frac{dx}{dt}$ | Model C4 $\frac{d^2x}{dt^2} = -\theta x - \delta\frac{dx}{dt} + \rho(z)$ |

Table 5.2: Glossaries of Dynamical Models Fitted to Data from dGp, with or without Discrepancy Model. The simulated data are obtained with the dGp A, B, and C with various values for $\delta$. The ODE models fitted against each dataset may be appropriately specified or misspecified. In both situations, we consider the augmented model with Random Fourier Features fitted using RFF-VAE.

### 5.5.1 Learning dynamics with RFF-VAE

#### 5.5.1.1 Free Fall Model

The ODE system generating the data (dGp A) for the Free Fall with air resistance model is given by :

$$\frac{d^2x}{dt^2} = -\theta - \delta\frac{dx}{dt}, \quad \text{(dGp A)}, \tag{5.14}$$

and this model is employed for simulating data. The simulated data, generated as $y(t) = x(t) + \mathcal{N}(0, \sigma^2)$, incorporates noise with a variance of $\sigma^2 = 0.1$. The parameters $\theta = 10$, with various values of $\delta$ within the time interval $[0, 10]$ are employed to fit the misspecified model denoted as

$$f_\theta(x, t) = -\theta.$$

In this case, the discrepancy is $\rho(z) = -\delta\frac{dx}{dt}$, where $z = \left(x, \frac{dx}{dt}\right)$.

Table A.3 gives the variational posterior estimates $(\mu_\phi; \sigma_\phi^2)$ across varying data size $n$, misspecification error $\delta$, and RFF size $D$ for model A2 fitted against dGp A. When $\delta = 0$, indicating the absence of model misspecification, the model accurately infers $\theta$ across varying RFF sizes $D$. Our approach tends to increase uncertainty, leading to larger variances in the variational posterior for $\sigma_\phi^2$. This suggests that the Gaussian process is not incorrectly updating the parameters, but rather adding uncertainty due to the increased complexity of the model. Nevertheless, it's important to note that the GP parameters are not equal to 0 in this situation. For example, when $D = 10$ with no misspecification $\delta = 0$, the variational posterior

obtained with RFF-VAE in Equation 5.11 for a dataset of size $n = 20$ are:

$$\mu_\theta = 10.026$$
$$\mu_\alpha = [-3.79, -0.42, -2.17, -3.40, 2.89, 3.71, -2.0119, 3.2983, 2.2707]$$
$$\sigma_\theta^2 = 0.003$$
$$\sigma_\alpha = [0.01, 0.01, 0.02, 0.02, 0.044, 0.008, 0.01, 0.03, 0.003, 0.003]$$

This observation suggests an identifiable problem that we will discuss in Section 5.5.2. Our findings show no significant robustness for inference against misspecification since we consistently underestimated the value of $\theta$, with the true parameter being 10, regardless of the misspecification level, data size, or RFF size. In Table A.3, the inference of $\theta$ is less biased without the RFF approximation ($D = 0$) across all error levels. In other words, when we account for misspecification with the RFF-augmented model, this leads to higher bias estimates for $\theta$.

In Figure 5.2, the predictive trajectories, obtained with the variational posterior mean $[\mu_\theta, \mu_\alpha]$ using RFF-VAE, are plotted alongside the trajectories obtained without the random Fourier expansion ($D = 0$), denoted as a VAE. These trajectories are compared against the simulated data using the Root Mean Square Error (RMSE). Our approach, RFF-VAE, exhibits superior RMSE performance when $\delta$ exceeds 0.1. This serves as evidence for the efficacy of the RFF approximation in accurately aligning with the true trajectory from the unknown dGp, with a misspecified dynamical model. Despite producing the wrong estimate for $\theta$, our approach correctly infers the misspecified dynamic trajectory.

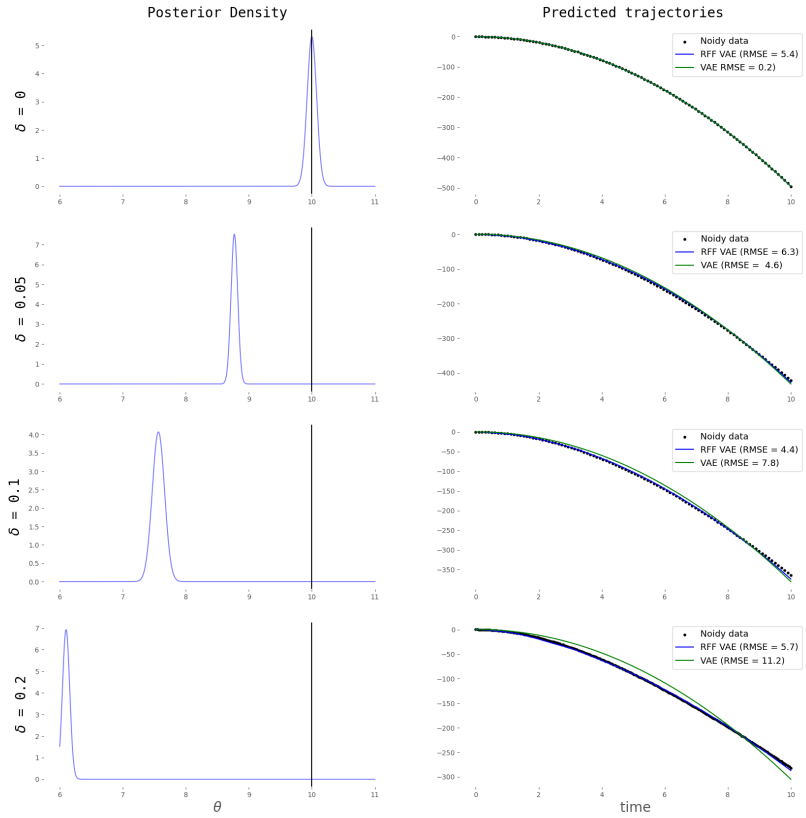



Figure 5.2: Misspecified Free Fall model - Left Panel: Variational Posterior Obtained with RFF-VAE (Model A2) with data coming from dGp A ($n = 20$) across varying error levels $\delta$ (top to bottom). Right Panel: Trajectories obtained with the variational posterior mean given on the left with $D = 100$ compared with the trajectories obtained with the VAE ($D = 0$). The observations are represented by black dot points.

#### 5.5.1.2 Pendulum Model

We use the classical pendulum model with air resistance (dGp B) as the ODE model for generating the data given by :

$$\frac{d^2x}{dt^2} = -\theta \sin(x) - \delta \frac{dx}{dt}, \quad \text{(dGp B)} \tag{5.15}$$

The simulated data are generated from this model with noise variance $\sigma^2 = 0.01$. The parameters are set as follows: $\theta = 10$, and $\delta$ takes values of 0, 0.05, 0.1, and 0.2, all within the time interval $[0, 10]$. We fit this data to two misspecified models, referred to as Model B2 and C2 in Table 5.2. The discrepancy is as $\rho(z) = -\delta \frac{dx}{dt}$, where $z = \left(x, \frac{dx}{dt}\right)$. However, model C2 has an additional error due to the absence of the sinusoidal driving force. The resulting variational posterior estimates $(\mu_\phi; \sigma^2_\phi)$ for model B2 and C2 are given respectively in Table A.4 and Table A.5. Model B2 exhibits inference robustness against misspecification with a correct estimate for $\theta$ even when $\delta = 0.2$ with high certainty (as observed already in Chapter 3). When employing the RFF-VAE approach, we notice a slight underestimation of $\theta$, accompanied by larger variational variances. Overall, integrating discrepancy does not enhance nor impede the correct inference of $\theta$ in Table A.4. This is not the case when using the misspecified model C2, where a large RFF size D for the approximation results in highly biased estimates and excessively large variances in Table A.4. This last model combines both types of misspecification (Type I and II) and fails to robustly learn $\theta$.

### 5.5.2 Identifiability

In situations where the true data-generating process significantly deviates from the class of models denoted as $\mathcal{Q}$, traditional mechanistic modelling faces substantial challenges, particularly concerning structural identifiabil-

ity ([Chis et al., 2011, Roberts, 2021, Curchoe, 2020]. Structural identifiability refers to the ability to find a unique and accurate value for each parameter in a model that can reproduce the observed data. When the model is misspecified or inadequately captures the complexities of the dGp, seeking structural identifiability can be counterproductive, as it may not yield meaningful parameter values. We use the same observed data from Equation 5.14 (dGp A), and fit them using the model A2 (Table 5.2) with RFF-VAE which is:

$$\frac{d^2x}{dt^2} = -\theta + \sum_{m=1}^{D} \alpha_m \Phi_m(z), \quad z = \left(x, \frac{dx}{dt}\right).$$

The true discrepancy is $-\theta - \delta\frac{dx}{dt}$ with $\theta = 10$ and $\delta = \{0, 0.1, 0.2\}$ whereas:

$$\begin{aligned}\mathbb{E}[\rho(z)] = \mathbb{E}[\sum_{m=1}^{D} \alpha_m \Phi_m(z)] &= \sum_{m=1}^{D} \mathbb{E}[\alpha_m \cos(a_m^\top z + b_m)] \\ &= \sum_{m=1}^{D} \mu_{\alpha_m} \cos\left(a_m^\top \begin{bmatrix} x \\ \frac{dx}{dt} \end{bmatrix} + b_m\right),\end{aligned}$$

with $\mu_{\alpha_m}$ the variational mean obtained for each latent parameter $\alpha_m$ via RFF-VAE, $a_m$ and $b_m$ the Random features components.

The dynamical model is misspecified with an unknown discrepancy proportional to the derivative state $\frac{dx}{dt}$. As shown in Figure 5.2, our approach provides biased estimates for $\theta$, but the resulting trajectories fit the true observations well.

Considering two-dimensional parameters $(x, \frac{dx}{dt})$, representing trajectory states and their derivatives, we can assess the GP discrepancy $\mathbb{E}[\rho(z)]$ and the unknown truth $-\delta\frac{dx}{dt}$. Figure 5.3 illustrates the correct right-hand side of the dGp and the right-hand side of model A2 across various errors using heatmaps, including their difference in the last column. The first column, $-\theta - \delta\frac{dx}{dt}$, corresponds to dGp A used to generate the observations with

different values for $\delta$. Since $\delta$ is close to zero, this primarily takes the value of $-\theta = -10$. In the second column, we have the discrepancy with the estimates obtained with RFF-VAE, given by $-\hat{\theta} + \mathbb{E}[\rho(z)]$. The last column assesses the difference between the other two columns, with the black line representing the true $x$ and $\frac{dx}{dt}$ used to fit the model. Essentially, the integral along that line tends to produce the correct answer on average, although it oscillates above and below the correct value of the right-hand side as the trajectory progresses. This behaviour may be attributed to the GP being trained on data with some level of noise, or due to the relative sparsity of training samples along that trajectory.

### 5.5.3 Enhancing RFF-VAE with Derivative State Information

#### Discrepancy model

We introduce a new discrepancy form by incorporating the derivative state component $\delta\frac{dx}{dt}$ so that the dynamical model with discrepancy becomes:

$$\frac{d^2x}{dt^2} = f_\theta(x,t) - \delta\frac{dx}{dt} + \rho(z), \tag{5.16}$$

where $z = \left(x, \frac{dx}{dt}\right)$.

We aim to jointly learn $\theta$, $\delta$, and the $\alpha_{1:D}$ parameters in Equation 5.7 using a variational bivariate Gaussian distribution $q(\theta, \delta, \alpha_{1:D} \mid x)$, following the methodology outlined in Section 5.4.1. This corresponds to Models A4, B4 and C4 in Table 5.2. If we remove the discrepancy term $\rho(\cdot)$, we only need to estimate the two (considered) independent dimensional parameters, $\theta$ and $\delta$, for Models A3, B3, and C3 as listed in Table 5.2. For this purpose, we adopt the same variational autoencoder framework as outlined in Chapter 3. Since the dGp is known in these experiments, we know that Model

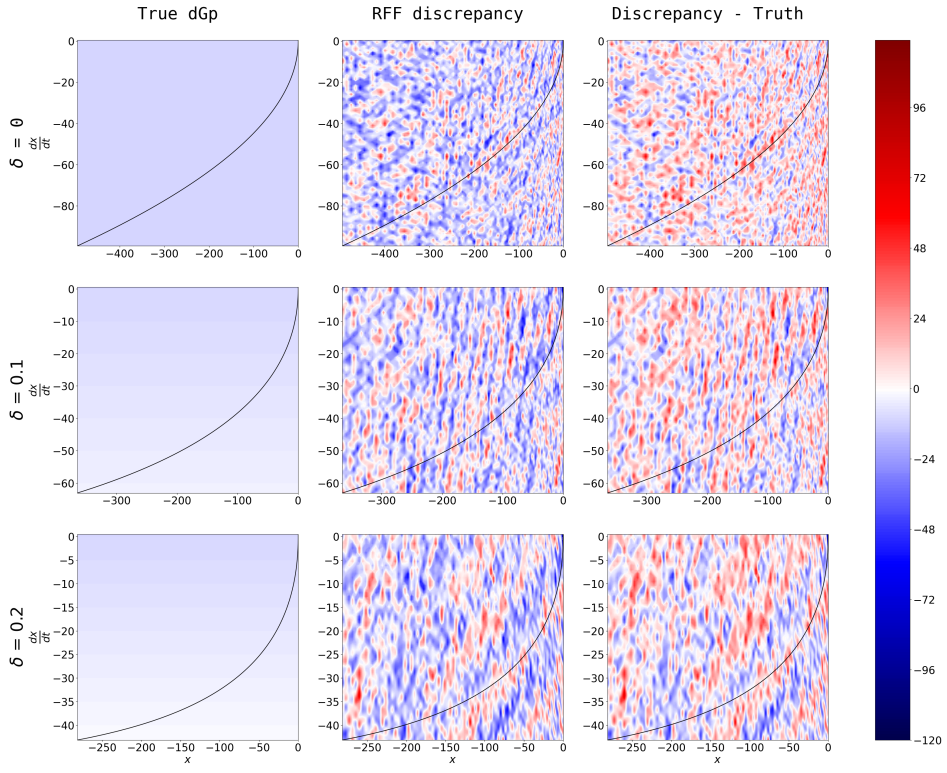



Figure 5.3: RFF learning effectiveness with data generated from dGp A (n=100) against the misspecified dynamical model A2 across several errors $\delta$. The true unknown dynamics, $-\theta - \delta\frac{dx}{dt}$, is represented in the heatmap on the left panel. the middle column displays the inferred RFF discrepancy, $-\theta - \rho(x, \frac{dx}{dt})$, with D = 100, and the last column shows the difference between them. The black line represents the model variables x and $\frac{dx}{dt}$ used in the training dataset.

A3 and B3 are correctly specified in this context.

**Data**

The data utilised remains consistent with the previous section, generated from the appropriate dynamics $f_\theta(x, t)$ outlined in Equation 5.14 (denoted as dGp A) and Equation 5.15 (referred to as dGp B or C), with the true parameter $\theta = 10$ and various values assigned to $\delta$.

**Results**

This approach is evaluated both with and without the inclusion of the discrepancy model $\rho(z)$. When $\rho(z)$ is not used in the model, the model is no longer misspecified. Figure 5.5 displays the contour plot of the variational bivariate distribution $q(\theta, \delta \mid x)$.

The blue contour plots in Figure 5.5 depict the variational posteriors without accounting for the discrepancy. Across models A3, B3, and C3, the parameter $\delta$ is largely overestimated, while $\theta$ tends to be slightly underestimated. This is attributed to the constraint introduced by selecting a variational family with independent parameters. In the simplest scenario where the data are generated with $\delta = 0$, it indicates that the VAE fails to accurately infer $\delta$, assigning a high value around 2.5. When compared with HMC with model A3, Figure 5.4 (left plot) reveals an underestimation of both $\theta$ and $\delta$, leading to suboptimal performance in inferring the parameter of interest $\theta$ with a value of 7.1. However, when $\delta = 0.2$, we correctly infer both parameters with HMC, unlike the VAE framework.

When incorporating discrepancy with the RFF-VAE approach (Model A4, B4, and C4), we observe higher uncertainty for both parameters $\theta$ and $\delta$, enabling a wider range of estimates for $\theta$ that can encompass the correct value ($\theta = 10$). Conversely, the parameter $\delta$ is consistently overestimated across all scenarios, with GP approximation. This suggests that the deriva-

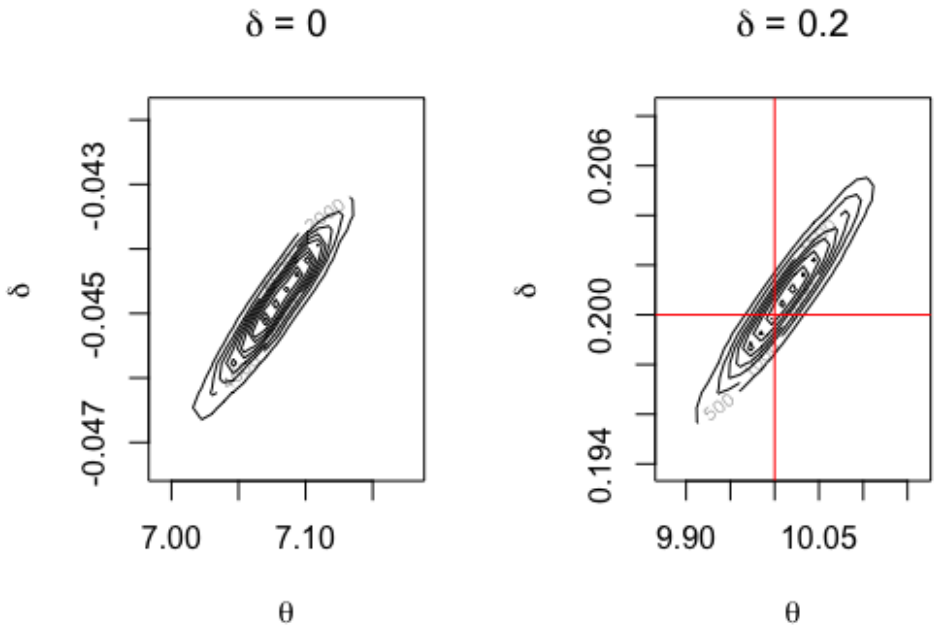



Figure 5.4: Variational posterior bivariate distribution $q(\theta, \delta \mid x)$ with HMC for Model A3 and data coming from the dGp A ($n = 100$). In the generated data, two values of $\delta$ are considered. On the left, the true value $\theta = 10$ and $\delta = 0$ is not accurately recovered. Conversely, on the right, both values are correctly inferred.

tive state plays a significant role in capturing dynamics within the generative model. We emphasise that the estimated variational variance with RFF-VAE is constantly notably higher than with HMC as shown in Figure 5.4. In dynamic models where we suspect some missing dynamics relying on the derivative state $\frac{dx}{dt}$, this boosting approach can be advantageous by providing better uncertainty quantification.

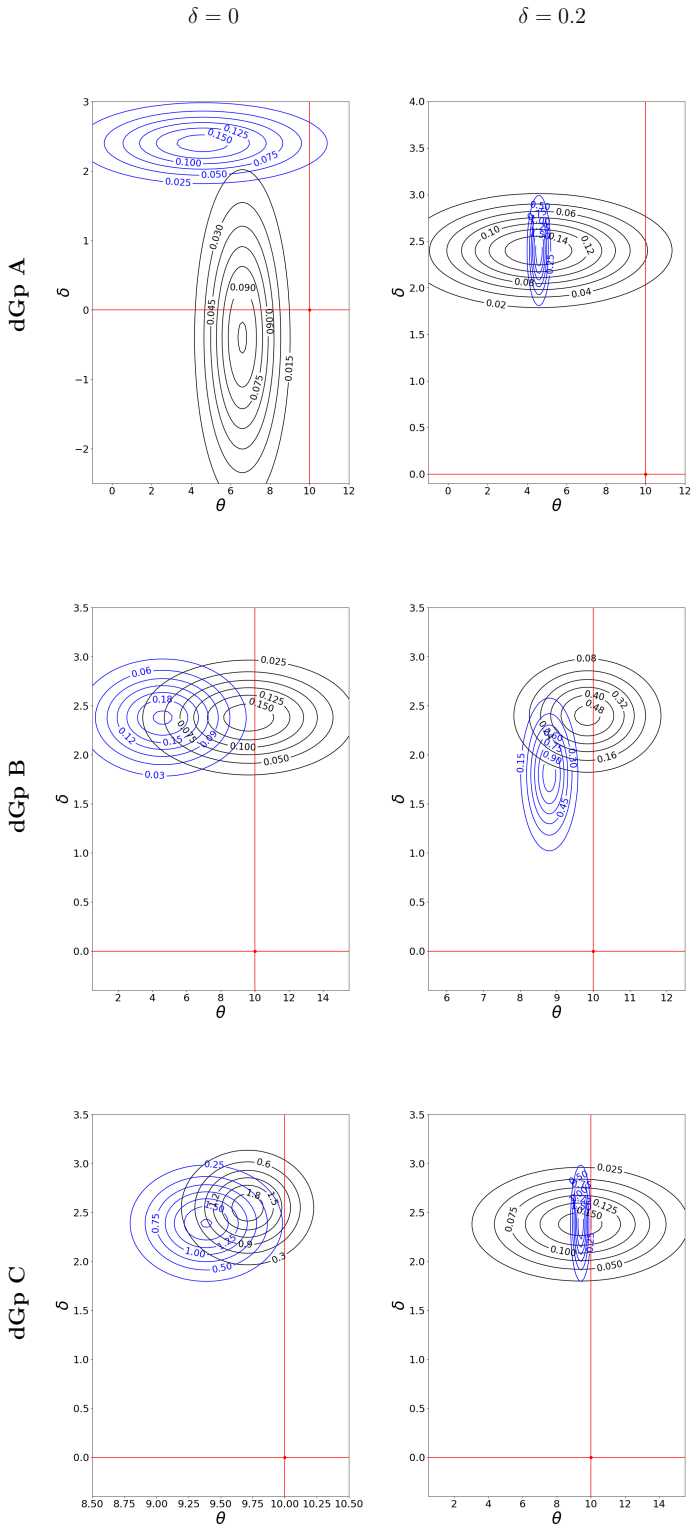



Figure 5.5: The variational posterior distribution of $\theta$ and $\delta$ is obtained using $n = 100$ observations from dGp A, B, and C (from top to bottom) fitted with models A3, B3, and C3 (depicted as blue contour plots) and fitted with discrepancy models A4, B4, and C4 (depicted as black contour plots) with $D = 200$ Random Fourier Features. The true values for $\theta$ and $\delta$ are indicated by the red lines.

## 5.6 Considerations

We have evaluated the inference and predictive trajectories across different dynamical models using our RFF-ODE framework. The findings indicate enhanced robustness to structural misspecification for predictions along with improved uncertainty quantification. However, we acknowledge several limitations:

1. Within the VAE framework, we adopt a mean-field family of independent parameters $\mathcal{Q}$ for the approximate distribution. This implies that all latent parameters, including those from the dynamic and in the RFF discrepancy, are assumed to be independent. As a result, this method inherently introduces a discrepancy between the true data-generating process and the approximation distribution. By contrast with HMC, the level of discrepancy $\delta$ is therefore consistently overestimated with RFF-VAE. Instead of approximating separate variables for each data point, we could consider Amortized Variational Inference (Ganguly et al. [2022]) where we assume that the local variational parameters can be predicted by a parameterised function of the data. Thus, once this function is estimated, the latent variables can be obtained by passing new data points through the function.

2. The computational time challenges outlined in Chapter 3 hinder the scalability of the inference method, as elaborated in the conclusion of this thesis. Specifically, we encountered limitations in applying our method to a particular misspecified scenario within cardiac physiology applications described in Example 3. Due to the excessive memory cost associated with the number of ODE steps required in backpropagation, an alternative approach is necessary.

3. A further limitation of our approach is that we have only considered

the radial basis function (RBF) kernel with fixed length-scale parameter equal to 1 for the Variational Fourier Features approximation. We intentionally chose the simplest kernel without conducting preliminary steps to identify the most effective one. While the RBF kernel is not an ideal choice, we would estimate the hyperparameters and explore alternative covariance functions, such as the Matérn kernel in future research. Nevertheless, the RBF kernel is suitable in this case, as ODEs generally have smooth right-hand sides, and the RBF kernel produces highly smooth, analytic functions with derivatives of all orders. It is not typical to have a discrepancy related to the first derivative. Usually, discrepancies are added to the output space, such as $y(t) = f(t) + \delta(t) + \epsilon$, as done in previous work. However, by considering a discrepancy in the first derivative, the aim is to correct the underlying equations and gain insights into the true dynamics.

## 5.7 Conclusion

When confronted with a presumed misspecified dynamical model, a common strategy is to shift towards data-driven models within machine learning. Several methods have been devised for dynamical systems governed by differential equations, aiming to construct predictive algorithms that effectively combine data and mechanistic prior knowledge. These techniques prove particularly valuable in scenarios involving imperfect data (Yang et al. [2020]) or inaccurately modeled biological mechanisms (Engelhardt et al. [2017]). These approaches acknowledge the limitations and uncertainties in the model structure, allowing for corrections to be made to better align with the true Data Generating Process. We introduced a novel grey-box framework for robust inference in misspecified dynamical systems combining differential programming techniques for Bayesian in-

ference with mechanistic knowledge associated with Gaussian process approximations. Our approach departs from previous studies by integrating Random Fourier features directly into the ODE solver within a variational autoencoder framework. When the mechanistic dynamical model is incorrectly specified, the discrepancy model learns the correct true observed trajectory, even though this may result in biased estimators for the meaningful parameters of the dynamical model. This can render the approach particularly useful for predictions in future research. Additionally, the discrepancy variational approach yields estimates with increased uncertainty, thereby enhancing uncertainty quantification when we suspect the dynamical model to be misspecified.

# Chapter 6

# Discussion

In this concluding section of the thesis, we will assess the contributions made and discuss some of the key remaining challenges.

## 6.1 Contributions of this thesis

If decision-makers are considering competing models, when should we expect them to drop their current model? What forms of misspecification are most likely to persist? Which inference procedure are more robust?
This thesis addresses these questions using various Bayesian variational methods within the context of model misspecification depicted in Figure 2.6. The thesis explores various methods and offers guidance on choosing the most suitable approach. Specifically, the thesis focuses on mechanistic models rooted in Ordinary Differential Equations (ODEs), which, to our knowledge, present a departure from typical examples previously examined in Variational Inference (VI) for handling model misspecification. Within this thesis, we advocate for grey-box modelling, a form of physics-based modelling, that bridges the gap between non-informative black-box models lacking theoretical knowledge and interpretability and white-box models relying solely on detailed but potentially misspecified physical principles.

- In Chapter 2, a comprehensive exploration is undertaken to elucidate the current limitations that standard Bayesian methods encounter when confronted with the challenges posed by model misspecification. This chapter also serves as a unifying platform, drawing together a wide spectrum of disciplines spanning Bayesian Statistics and Machine Learning, coalescing them into a coherent and interconnected framework. We emphasise how robustness in inference is often viewed through the prism of contamination rather than structural error, which is the focal point of interest in this thesis.

- In Chapter 3, we have harnessed the power of automatic differentiation variational inference by innovatively blending a VAE architecture and differentiable gradient in dynamical ODE systems from simulated data. We discussed the challenges associated with employing traditional adjoint and sensitivity methods in conjunction with automatic differentiation for fitting ODE-based models. This chapter represents one step towards improving the efficiency and effectiveness of inference processes within mechanistic modelling by integrating physic-informed models.

- In Chapter 4, our primary focus has been the computation of generalised posteriors using a state-of-the-art generalised variational inference approach, with a careful selection of robust loss functions. This chapter is a comprehensive effort to robustify Bayesian statistical inference methods and make them more resilient against model misspecification.

- Chapter 5 presents an innovative model augmentation approach that addresses discrepancies in misspecified models through the application of Variational Inference with Random Fourier Features. Our approach is novel as it integrates the discrepancy model within the

ODE model, enabling backpropagation through the Gaussian Process approximation to obtain posterior distributions.

## 6.2 Open problems

**Computational Challenges for ODEs backpropagation**

Throughout this research project, we encountered several computational challenges while employing automatic differentiation for variational inference. In particular, utilising backpropagation through a mechanistic dynamical model is a recent practice. The foundational work by Ricky Chen et al. [2018] primarily emphasises the replacement of dynamics with neural networks. We faced challenges with the slower performance of this method in PyTorch, encountering memory-intensive issues, when dealing with ODE-based models. This challenge became particularly pronounced as we endeavored to tackle complex real-world models involving non-linearity and stiffness ODE models in cardiac physiology. Exploring forward-mode Automatic Differentiation in PyTorch (currently in beta at pytorch.org/tutorials/intermediate/forward_ad_usage.html) instead of relying on reverse-mode backpropagation, though not pursued due to time constraints, could be a more efficient approach for accelerating and scaling the methods.

**Robust Inference in the Presence of Structural Model Misspecification**

Achieving robustness for misspecified mechanistic models involves ensuring that a modelling approach remains resilient and effective, even when the assumed mechanistic model does not align with the true underlying dynamics of the system. Robustness in this context implies that the modelling framework can still provide meaningful and reliable results in real-world situations, even though achieving robustness may come at a cost. Exist-

ing robustness measures commonly used in the literature seem inadequate for determining whether inference remains sufficiently efficient under structural misspecification. We believe there is significant potential for research in mechanistic models in this regard.

**GVI for Dynamical Models: Assessing its Real Benefits**

Generalising Bayesian inference introduces challenges in choosing from various alternatives. Unlike model selection, which revolves around choosing a statistical model for the data, the generalised framework lacks a theoretical recipe for selecting a specific loss function. In complex data scenarios, determining feasible specifications for $\ell$, $\theta$, and $\Pi$ becomes challenging due to uncertainties in quantifying distortion or corruption in the data, making it difficult to choose between non-likelihood-based and likelihood-based loss functions. Do we take a risk by switching to a more generalised framework? While the choices made in this thesis have been subjective, they serve as benchmarks rather than definitive answers, highlighting concerns about the reliability of the generalised Bayesian inference framework. Notably, they have shown varying degrees of effectiveness, ranging from limited utility to virtually non-existent performance when dealing with structural errors in the considered ODE models. We chose to rely on the recent generalised variational inference paradigm introduced in recent literature, by the core paper by Knoblauch et al. [2019]. To the best of our knowledge, no papers have yet utilised this paradigm for ODE-based models, making it an ambitious first step.

# Bibliography

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

P. M. Abeyasinghe, D. R. de Paula, S. Khajehabdollahi, S. R. Valluri, A. M. Owen, and A. Soddu. Role of Dimensionality in Predicting the Spontaneous Behavior of the Brain Using the Classical Ising Model and the Ising Model Implemented on a Structural Connectome. *Brain Connect.*, 8(7):444–455, Sept. 2018.

C. Agostinelli and L. Greco. A weighted strategy to handle likelihood uncertainty in Bayesian inference. *Comput. Stat.*, 28(1):319–339, Feb. 2013.

P. Alquier and J. Ridgway. Concentration of tempered posteriors and of their variational approximations. June 2017.

P. Alquier, J. Ridgway, and N. Chopin. On the properties of variational

approximations of Gibbs posteriors. *J. Mach. Learn. Res.*, 17(236):1–41, 2016.

M. Altamirano, F.-X. Briol, and J. Knoblauch. Robust and Scalable Bayesian Online Changepoint Detection. *arXiv*, Feb. 2023.

I. Andrianakis, I. R. Vernon, N. McCreesh, T. J. McKinley, J. E. Oakley, R. N. Nsubuga, M. Goldstein, and R. G. White. Bayesian history matching of complex infectious disease models using emulation: a tutorial and a case study on HIV in Uganda. *PLoS Comput. Biol.*, 11(1):e1003968, Jan. 2015.

E. Angelino, M. J. Johnson, and R. P. Adams. Patterns of Scalable Bayesian Inference. *arXiv*, Feb. 2016.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *arXiv*, Jan. 2017.

R. Astudillo and P. I. Frazier. Thinking Inside the Box: A Tutorial on Grey-Box Bayesian Optimization. In *2021 Winter Simulation Conference (WSC)*, pages 1–15. IEEE, Dec. 2021.

A. Barp, F.-X. Briol, A. B. Duncan, M. Girolami, and L. Mackey. Minimum Stein Discrepancy Estimators. June 2019.

A. Basu, I. R. Harris, N. L. Hjort, and M. C. Jones. Robust and Efficient Estimation by Minimising a Density Power Divergence. *Biometrika*, 85 (3):549–559, 1998.

M. A. Beaumont, W. Zhang, and D. J. Balding. Approximate Bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, Dec. 2002.

M. Betancourt. A Conceptual Introduction to Hamiltonian Monte Carlo. Jan. 2017.

M. C. Bishop. Pattern Recognition and Machine Learning. 2006.

P. G. Bissiri, C. C. Holmes, and S. G. Walker. A general framework for updating belief distributions. *J. R. Stat. Soc. Series B Stat. Methodol.*, 78(5):1103–1130, Nov. 2016.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational Inference: A Review for Statisticians. Jan. 2016.

M. Blondel, Q. Berthet, M. Cuturi, R. Frostig, S. Hoyer, F. Llinares-López, F. Pedregosa, and J.-P. Vert. Efficient and Modular Implicit Differentiation. *arXiv*, May 2021.

G. E. P. Box and N. R. Draper. Empirical model-building and response surfaces. *Wiley series in probability and mathematical statistics.*, 669, 1987.

J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

F.-X. Briol, A. Barp, A. B. Duncan, and M. Girolami. Statistical Inference for Generative Models with Maximum Mean Discrepancy. June 2019.

T. Broderick, R. Giordano, and R. Meager. An automatic Finite-Sample robustness metric: When can dropping a little data make a big difference? Technical Report 2011.14999, July 2023.

S. L. Brunton, J. L. Proctor, and J. Nathan Kutz. Discovering governing equations from data: Sparse identification of nonlinear dynamical systems. sep 2015.

Y. Burda, R. Grosse, and R. Salakhutdinov. Importance Weighted Autoencoders. *arXiv*, Sept. 2015.

B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 6572–6583, Red Hook, NY, USA, Dec. 2018. Curran Associates Inc.

B.-E. Chérief-Abdellatif and P. Alquier. Finite sample properties of parametric MMD estimation: robustness to misspecification and dependence. *arXiv*, Dec. 2019.

H. Chernoff. A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the sum of Observations. *aoms*, 23(4):493–507, Dec. 1952.

O.-T. Chis, J. R. Banga, and E. Balsa-Canto. Structural identifiability of systems biology models: a critical comparison of methods. *PLoS One*, 6 (11):e27755, Nov. 2011.

A. Cichocki and S.-I. Amari. Families of alpha- beta- and gamma- divergences: Flexible and robust measures of similarities. *Entropy*, 12(6): 1532–1568, June 2010.

S. Conti, J. P. Gosling, and J. E. Oakley. Gaussian process emulation of dynamic computer codes. *Biometrika*, 96(3):663–676, Sept. 2009.

S. Coveney, C. H. Roney, C. Corrado, R. D. Wilkinson, J. E. Oakley, S. A. Niederer, and R. H. Clayton. Calibrating cardiac electrophysiology models using latent Gaussian processes on atrial manifolds. *Sci. Rep.*, 12(1):16572, Oct. 2022.

P. S. Craig, M. Goldstein, A. H. Seheult, and J. A. Smith. Pressure Matching for Hydrocarbon Reservoirs: A Case Study in the Use of Bayes Linear

Strategies for Large Computer Experiments. In *Case Studies in Bayesian Statistics*, pages 37–93. Springer New York, 1997.

K. Csilléry, M. G. B. Blum, O. E. Gaggiotti, and O. François. Approximate Bayesian Computation (ABC) in practice. *Trends Ecol. Evol.*, 25(7):410–418, July 2010.

I. Csiszar. $I$-divergence geometry of probability distributions and minimization problems. *Ann. Probab.*, 3(1):146–158, Feb. 1975.

Y. L. Cun. A Theoretical Framework for Back-Propagation. *arXiv*, 1988.

C. L. Curchoe. All Models Are Wrong, but Some Are Useful. *J. Assist. Reprod. Genet.*, 37(10):2389–2391, Oct. 2020.

P. Dawid, M. Musio, and L. Ventura. Minimum scoring rule inference. Mar. 2014.

R. de Heide, A. Kirichenko, N. Mehta, and P. Grünwald. Safe-Bayesian generalized linear regression. *arXiv*, Oct. 2019.

C. Dellaporta, J. Knoblauch, T. Damoulas, and F.-X. Briol. Robust bayesian inference for simulator-based models via the MMD posterior bootstrap. *arXiv*, 151:943–970, 2022.

P. Diaconis and D. Freedman. On the Consistency of Bayes Estimates. *Ann. Stat.*, 14(1):1–26, 1986.

C. Doersch. Tutorial on Variational Autoencoders. *arXiv*, June 2016.

M. D. Donsker and S. R. S. Varadhan. Asymptotic evaluation of certain Markov process expectations for large time - III. *Commun. Pure Appl. Math.*, 29(4):389–461, July 1976.

J. Duchi, E. Hazan, and Y. Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.*, 12 (61):2121–2159, 2011.

L. Duncker, G. Bohner, J. Boussard, and M. Sahani. Learning interpretable continuous-time models of latent stochastic dynamical systems. Feb. 2019.

S. Eguchi. A differential geometric approach to statistical inference on the basis of contrast functionals. *hmj*, 15(2):341–391, Jan. 1985.

B. Engelhardt, M. Kschischo, and H. Fröhlich. A Bayesian approach to estimating hidden variables as well as missing and wrong molecular interactions in ordinary differential equation-based mathematical models. *J. R. Soc. Interface*, 14(131), June 2017.

D. T. Frazier. Robust and Efficient Approximate Bayesian Computation: A Minimum Distance Approach. June 2020.

D. T. Frazier, C. P. Robert, and J. Rousseau. Model Misspecification in ABC: Consequences and Diagnostics. *arXiv*, Aug. 2017.

H. Fujisawa and S. Eguchi. Robust parameter estimation with a small bias against heavy contamination. *J. Multivar. Anal.*, 99(9):2053–2081, Oct. 2008.

P. Gahungu, C. W. Lanyon, M. A. Alvarez, E. Bainomugisha, M. Smith, and R. D. Wilkinson. Adjoint-aided inference of Gaussian process driven differential equations. Feb. 2022.

A. Ganguly, S. Jain, and U. Watchareeruetai. Amortized Variational Inference: A Systematic Review. Sept. 2022.

M. L. Garsdal, V. Søgaard, and S. M. Sørensen. Generative time series models using Neural ODE in Variational Autoencoders. *arXiv*, Jan. 2022.

A. Gelman, A. Vehtari, D. Simpson, C. C. Margossian, B. Carpenter, Y. Yao, L. Kennedy, J. Gabry, P.-C. Bürkner, and M. Modrák. Bayesian workflow. Nov. 2020.

I. M. Gherman, L. Marucci, Z. Abdallah, C. Grierson, T. Gorochowski, and W. Pang. Bridging the gap between mechanistic biological models and machine learning surrogates. Sept. 2022.

S. Ghosal, J. K. Ghosh, and A. W. van der Vaart. Convergence Rates of Posterior Distributions. *Ann. Stat.*, 28(2):500–531, 2000.

M. Goldstein and J. Rougier. Reified Bayesian modelling and inference for physical systems. *J. Stat. Plan. Inference*, 139(3):1221–1239, Mar. 2009.

N. S. Gorbach, S. Bauer, and J. M. Buhmann. Scalable Variational Inference for Dynamical Systems. *arXiv [stat.ML]*, May 2017.

L. Greco, W. Racugno, and L. Ventura. Robust likelihood functions in Bayesian inference. *J. Stat. Plan. Inference*, 138(5):1258–1270, May 2008.

A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola. A kernel method for the two-sample-problem. *Adv. Neural Inf. Process. Syst.*, pages 513–520, Dec. 2006.

A. Griewank and A. Walther. *Evaluating Derivatives*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, Jan. 2008.

P. Grünwald. Contextuality of misspecification and Data-Dependent losses. *SSO Schweiz. Monatsschr. Zahnheilkd.*, 31(4):495–498, Nov. 2016.

P. Grünwald and T. van Ommen. Inconsistency of Bayesian Inference for Misspecified Linear Models, and a Proposal for Repairing It. Dec. 2014.

W. K. Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1):97–109, 1970.

J. Hensman, N. Durrande, and A. Solin. Variational Fourier Features for Gaussian Processes. *J. Mach. Learn. Res.*, 18(151):1–52, 2018.

D. Higdon, M. Kennedy, J. C. Cavendish, J. A. Cafeo, and R. D. Ryne. Combining Field Data and Computer Simulations for Calibration and Prediction. *SIAM J. Sci. Comput.*, 26(2):448–466, Jan. 2004.

M. D. Hoffman and D. M. Blei. Structured Stochastic Variational Inference. Apr. 2014.

M. D. Hoffman and A. Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. Nov. 2011.

M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic Variational Inference. *J. Mach. Learn. Res.*, 14:1303–1347, 2013.

P. J. Huber. Robust Statistics. In M. Lovric, editor, *International Encyclopedia of Statistical Science*, pages 1248–1251. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

H. Husain and J. Knoblauch. Adversarial Interpretation of Bayesian Inference. In S. Dasgupta and N. Haghtalab, editors, *Proceedings of The 33rd International Conference on Algorithmic Learning Theory*, volume 167 of *Proceedings of Machine Learning Research*, pages 553–572. PMLR, 2022.

A. Hyvarinen. Estimation of Non-Normalized Statistical Models by Score Matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

J. Jewson, J. Q. Smith, and C. Holmes. Principles of Bayesian Inference using General Divergence Criteria. Feb. 2018.

J. Jewson, J. Q. Smith, and C. Holmes. On the Stability of General Bayesian Inference. Jan. 2023.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An Introduction to Variational Methods for Graphical Models. *Mach. Learn.*, 37 (2):183–233, Nov. 1999.

M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *J. R. Stat. Soc. Series B Stat. Methodol.*, 63(3):425–464, 2001.

P. Kidger, R. T. Q. Chen, and T. Lyons. "Hey, that's not an ODE": Faster ODE Adjoints via Seminorms. Sept. 2020.

D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. Dec. 2014.

D. P. Kingma and M. Welling. Auto-Encoding variational bayes. Dec. 2013.

J. Knoblauch. Robust Deep Gaussian Processes. Apr. 2019.

J. Knoblauch, J. Jewson, and T. Damoulas. Doubly Robust Bayesian Inference for Non-Stationary Streaming Data with $\beta$-Divergences. *arXiv*, June 2018.

J. Knoblauch, J. Jewson, and T. Damoulas. Generalized Variational Inference: Three arguments for deriving new Posteriors. Apr. 2019.

I. Krissaane, K. Hampton, J. Alshenaifi, and R. Wilkinson. Anomaly detection Semi-Supervised framework for sepsis treatment. In *2019 Computing in Cardiology (CinC)*, pages Page 1–Page 4, Sept. 2019.

N. R. Kristensen, H. Madsen, and S. B. Jørgensen. Parameter estimation in stochastic grey-box models. *Automatica*, 40(2):225–237, Feb. 2004.

S. Kullback and R. A. Leibler. On Information and Sufficiency. *aoms*, 22 (1):79–86, Mar. 1951.

S. Kurtek and K. Bharath. Bayes sensitivity with Fisher-Rao metric. Mar. 2014.

S. Kurtek and K. Bharath. Bayesian sensitivity analysis with the Fisher–Rao metric. *Biometrika*, 102(3):601–616, Sept. 2015.

T. Laisk, M. Lepamets, M. Koel, E. Abner, Estonian Biobank Research Team, and R. Mägi. Genome-wide association study identifies five risk loci for pernicious anemia. *Nat. Commun.*, 12(1):3761, June 2021.

C. L. Lei, S. Ghosh, D. G. Whittaker, Y. Aboelkassem, K. A. Beattie, C. D. Cantwell, T. Delhaas, C. Houston, G. M. Novaes, A. V. Panfilov, P. Pathmanathan, M. Riabiz, R. W. Dos Santos, J. Walmsley, K. Worden, G. R. Mirams, and R. D. Wilkinson. Considering discrepancy when calibrating a mechanistic electrophysiology model. *Philos. Trans. A Math. Phys. Eng. Sci.*, 378(2173):20190349, June 2020.

M. Lienen and S. Günnemann. torchode: A Parallel ODE Solver for PyTorch. Oct. 2022.

O. Linial, N. Ravid, D. Eytan, and U. Shalit. Generative ODE Modeling with Known Unknowns. Mar. 2020.

J. Lintusaari, M. U. Gutmann, R. Dutta, S. Kaski, and J. Corander. Fundamentals and Recent Developments in Approximate Bayesian Computation. *Syst. Biol.*, 66(1):e66–e82, Jan. 2017.

B. Macdonald and D. Husmeier. Gradient Matching Methods for Computational Inference in Mechanistic Models for Systems Biology: A Review and Comparative Analysis. *Front Bioeng Biotechnol*, 3:180, Nov. 2015.

R. L. Magnusson. Huber-white robust estimators for linear regression models. *Journal of the American Statistical Association*, 1975. doi: 10.1080/01621459.1975.10480232.

S. Mandt, J. McInerney, F. Abrol, R. Ranganath, and D. Blei. Variational Tempering. Nov. 2014.

J.-M. Marin, P. Pudlo, C. P. Robert, and R. Ryder. Approximate Bayesian Computational methods. Jan. 2011.

S. Massaroli, M. Poli, J. Park, A. Yamashita, and H. Asama. Dissecting Neural ODEs. Feb. 2020.

I. Matei, J. de Kleer, C. Somarakis, R. Rai, and J. S. Baras. Interpretable machine learning models: a physics-based view. Mar. 2020.

T. Matsubara, J. Knoblauch, F.-X. Briol, and C. J. Oates. Robust Generalised Bayesian Inference for Intractable Likelihoods. *arXiv*, Apr. 2021.

T. Matsubara, J. Knoblauch, F.-X. Briol, and C. J. Oates. Generalised Bayesian Inference for Discrete Intractable Likelihood. June 2022.

M. A. Medina, J. L. M. Olea, C. Rush, and A. Velez. On the Robustness to Misspecification of $\alpha$-posteriors and Their Variational Approximations. *J. Mach. Learn. Res.*, 23(147):1–51, 2022.

A. C. Miller, N. J. Foti, A. D'Amour, and R. P. Adams. Reducing Reparameterization Gradient Variance. May 2017.

J. W. Miller and D. B. Dunson. Robust Bayesian inference via coarsening. *J. Am. Stat. Assoc.*, 114(527):1113–1125, 2019.

G. R. Mirams, P. Pathmanathan, R. A. Gray, P. Challenor, and R. H. Clayton. Uncertainty and variability in computational and mathematical models of cardiac physiology. *J. Physiol.*, 594(23):6833–6847, Dec. 2016.

A. Moreno, T. Adel, E. Meeds, J. M. Rehg, and M. Welling. Automatic Variational ABC. June 2016.

J. M. Murphy, B. B. B. Booth, M. Collins, G. R. Harris, D. M. H. Sexton, and M. J. Webb. A methodology for probabilistic predictions of regional climate change from perturbed physics ensembles. *Philos. Trans. A Math. Phys. Eng. Sci.*, 365(1857):1993–2028, Aug. 2007.

M. Niu, S. Rogers, M. Filippone, and D. Husmeier. Fast Parameter Inference in Nonlinear Dynamical Systems using Iterative Gradient Matching. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1699–1707, New York, New York, USA, 2016. PMLR.

J. E. Oakley and B. D. Youngman. Calibration of Stochastic Computer Simulators Using Likelihood Emulation. *Technometrics*, 59(1):80–92, Jan. 2017.

M. Opper and D. Saad. *Advanced Mean Field Methods: Theory and Practice*. MIT Press, 2001.

J. T. Ormerod and M. P. Wand. Explaining Variational Approximations. *Am. Stat.*, 64(2):140–153, May 2010.

J. T. Ormerod and M. P. Wand. Gaussian Variational Approximate Inference for Generalized Linear Mixed Models. *J. Comput. Graph. Stat.*, 21 (1):2–17, Jan. 2012.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library.

D. Pati, A. Bhattacharya, and Y. Yang. On Statistical Optimality of Variational Bayes. Dec. 2017.

F. Peng and D. K. Dey. Bayesian Analysis of Outlier Problems Using Divergence Measures. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 23(2):199–213, 1995.

M. Poli and S. Massaroli. TorchDyn: Implicit models and neural numerical methods in PyTorch. https://physical-reasoning.github.io/assets/pdf/papers/03.pdf. Accessed: 2024-3-7.

L. S. Pontryagin. *The Mathematical Theory of Optimal Processes*. Interscience Publishers, 1962.

C. A. Pope, J. P. Gosling, S. Barber, J. S. Johnson, T. Yamaguchi, G. Feingold, and P. G. Blackwell. Gaussian Process Modeling of Heterogeneity and Discontinuities Using Voronoi Tessellations. *Technometrics*, 63(1): 53–63, Jan. 2021.

J. W. Priddle, S. A. Sisson, D. T. Frazier, and C. Drovandi. Efficient Bayesian synthetic likelihood with whitening transformations. Sept. 2019.

Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin. Variational Autoencoder for Deep Learning of Images, Labels and Captions. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett,

editors, *Advances in Neural Information Processing Systems 29*, pages 2352–2360. Curran Associates, Inc., 2016.

A. Rahimi and B. Recht. Random features for Large-Scale kernel machines. *Adv. Neural Inf. Process. Syst.*, 2007.

J. O. Ramsay, G. Hooker, C. Cao, and others. Estimating differential equations. *Preprint, Department of*, 2005.

R. Ranganath, S. Gerrish, and D. M. Blei. Black box variational inference. Dec. 2013.

A. Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, volume 4.1, pages 547–562. University of California Press, Jan. 1961.

D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. May 2015.

H. Robbins and S. Monro. A Stochastic Approximation Method. *Ann. Math. Stat.*, 22(3):400–407, Sept. 1951.

D. A. Roberts. Why is AI hard and Physics simple? Mar. 2021.

G. Roeder, P. Grant, A. Phillips, N. Dalchau, and T. Meeds. Efficient Amortised Bayesian Inference for Hierarchical and Nonlinear Dynamical Systems. June 2019.

E. Ronchetti. Robust inference by influence functions. *J. Stat. Plan. Inference*, 57(1):59–72, Jan. 1997.

J. Rudi, J. Bessac, and A. Lenzi. Parameter Estimation with Dense and Convolutional Neural Networks Applied to the FitzHugh-Nagumo ODE. Dec. 2020.

W. Rudin. *Fourier Analysis on Groups — Wiley*. Sept. 2011.

T. Ryder, A. Golightly, A. Stephen McGough, and D. Prangle. Black-box Variational Inference for Stochastic Differential Equations. Feb. 2018.

S. M. Schmon, P. W. Cannon, and J. Knoblauch. Generalized Posteriors in Approximate Bayesian Computation. Nov. 2020.

M. Schober, D. K. Duvenaud, and P. Hennig. Probabilistic ODE solvers with Runge-Kutta means. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 739–747. Curran Associates, Inc., 2014.

B. Schölkopf. Causality for machine learning. Nov. 2019.

B. Sengupta, K. J. Friston, and W. D. Penny. Efficient gradient computation for dynamical models. *Neuroimage*, 98:521–527, Sept. 2014.

R. Serban and A. C. Hindmarsh. CVODES: The Sensitivity-Enabled ODE solver in SUNDIALS. *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 257–269, June 2008.

H. Shin and M. Choi. Physics-informed variational inference for uncertainty quantification of stochastic differential equations. *J. Comput. Phys.*, 487: 112183, Aug. 2023.

SI Amari, O. E. Barndorff-Nielsen, R. E. Kass, S. L. Lauritzen, and C. R. Rao. Differential geometry in statistical inference. *Lect. Notes Monogr. Ser.*, 10:i–240, 1987.

S. A. Sisson, Y. Fan, and M. A. Beaumont. Overview of Approximate Bayesian Computation. Feb. 2018.

S. Sivaganesan. Global and Local Robustness Approaches: Uses and Limitations. In D. R. Insua and F. Ruggeri, editors, *Robust Bayesian Analysis*, pages 89–108. Springer New York, New York, NY, 2000.

A. F. M. Smith and J. M. Bernardo. *Bayesian Theory*. Wiley & Sons, Limited, John, 2008.

B. Speelpenning. Compiling fast partial derivatives of functions given by algorithms. Technical report, Jan. 1980.

C. Stein. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. 1972.

D. Tran, R. Ranganath, and D. M. Blei. Hierarchical Implicit Models and Likelihood-Free Variational Inference. Feb. 2017.

M. Tschannen, O. Bachem, and M. Lucic. Recent Advances in Autoencoder-Based Representation Learning. Dec. 2018.

H. J. A. F. Tulleken. Grey-box modelling and identification using physical knowledge and bayesian techniques. *Automatica*, 29(2):285–308, Mar. 1993.

A. W. van der Vaart. *Asymptotic Statistics*. Cambridge University Press, Oct. 1998.

V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and Composing Robust Features with Denoising Autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1096–1103, New York, NY, USA, 2008. ACM.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *FNT in Machine Learning*, 1(1–2):1–305, 2007.

S. G. Walker. Bayesian inference with misspecified models. *J. Stat. Plan. Inference*, 143(10):1621–1633, Oct. 2013.

B. Wang and D. Titterington. Convergence and asymptotic normality of variational Bayesian approximations for exponential family models with missing values. July 2012.

B. Wang and D. M. Titterington. Inadequacy of interval estimates corresponding to variational Bayesian approximations. In R. G. Cowell and Z. Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, volume R5 of *Proceedings of Machine Learning Research*, pages 373–380. PMLR, 2005.

Y. Wang and D. M. Blei. Frequentist Consistency of Variational Bayes. May 2017.

Y. Wang and D. M. Blei. Variational Bayes under Model Misspecification. May 2019.

R. D. Wilkinson. Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. *Stat. Appl. Genet. Mol. Biol.*, 12(2):129–141, May 2013.

R. D. Wilkinson, M. Vrettas, D. Cornford, and J. E. Oakley. Quantifying simulator discrepancy in Discrete-Time dynamical simulators. *J. Agric. Biol. Environ. Stat.*, 16(4):554–570, 2011.

D. Williamson, M. Goldstein, L. Allison, A. Blaker, P. Challenor, L. Jackson, and K. Yamazaki. History matching for exploring and reducing climate model parameter space using observations and a large perturbed physics ensemble. *Clim. Dyn.*, 41(7):1703–1729, Oct. 2013.

Y. Yang, M. Aziz Bhouri, and P. Perdikaris. Bayesian differential program-

ming for robust systems identification under uncertainty. *Proc. Math. Phys. Eng. Sci.*, 476(2243):20200290, Nov. 2020.

Ç. Yıldız, M. Heinonen, and H. Lähdesmäki. ODE$^2$VAE: Deep generative second order ODEs with Bayesian neural networks. May 2019.

C. You, J. T. Ormerod, and S. Müller. On variational Bayes estimation and variational information criteria for linear regression models. *Aust. N. Z. J. Stat.*, 56(1):73–87, Mar. 2014.

C. Zhang, J. Butepage, H. Kjellstrom, and S. Mandt. Advances in Variational Inference. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):2008–2026, Aug. 2019.

F. Zhang and C. Gao. Convergence Rates of Variational Posterior Distributions. Dec. 2017.

S. Zhao, J. Song, and S. Ermon. InfoVAE: Balancing Learning and Inference in Variational Autoencoders. *AAAI*, 33(01):5885–5892, July 2019.

Y. Zhou, Q. Zhou, and H. Wang. Inferring the unknown parameters in differential equation by gaussian process regression with constraint. *Computational and Applied Mathematics*, 41(6):280, Aug. 2022.

# Appendices

# Appendix A

# Supplementary materials

## A.1 Chapter 3

### A.1.1 Well specified scenarios

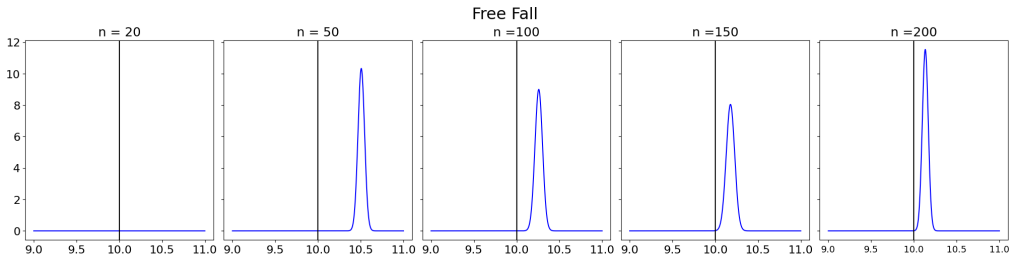
| | n=20 | | n=50 | | n=100 | | n=150 | | n=200 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\bar{\mu}$ | $\bar{\sigma}^2$ | $\bar{\mu}$ | $\bar{\sigma}^2$ | $\bar{\mu}$ | $\bar{\sigma}^2$ | $\bar{\mu}$ | $\bar{\sigma}^2$ | $\bar{\mu}$ | $\bar{\sigma}^2$ |
| Free Fall | 10.007 | 0.003 | 10.007 | 0.001 | 9.99 | 0.002 | 10.079 | 0.003 | 10.015 | 0.002 |
| Simple harmonic motion | 9.954 | 0.003 | 9.996 | 0.004 | 9.993 | 0.003 | 9.923 | 0.003 | 10.029 | 0.005 |

Table A.1: Well-specified models. The Variational posterior distributions for the free fall and simple harmonic motion models are depicted respectively in Figure 3.8a and 3.8b.

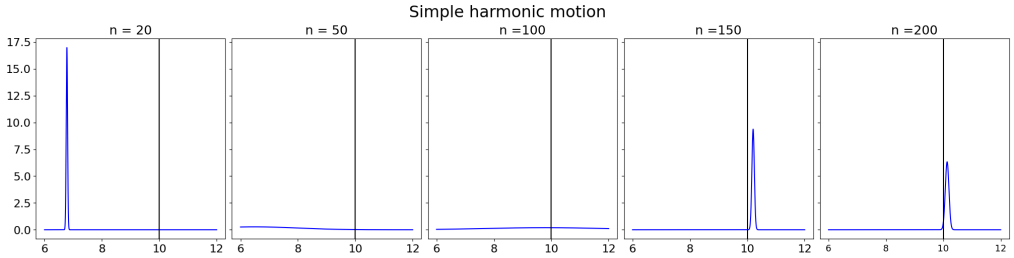
### A.1.2 Misspecified scenarios

| | $\delta = 0.05$ | | $\delta = 0.1$ | | $\delta = 0.2$ | |
|---|---|---|---|---|---|---|
| | $\bar{\mu}$ | $\bar{\sigma}^2$ | $\bar{\mu}$ | $\bar{\sigma}^2$ | $\bar{\mu}$ | $\bar{\sigma}^2$ |
| Free Fall | 8.760 | 0.002 | 7.734 | 0.002 | 6.203 | 0.003 |
| Simple harmonic motion | 8.075 | 3.7 | 8.067 | 0.065 | 6.487 | 0.006 |

Table A.2: Misspecified scenarios. Variational posterior for the free fall and the simple harmonic motion models for $n = 100$.




(a)




(b)

Figure A.1: Variational posterior distribution for each model (a) Model 3.4.1, (b) Model 3.4.3. We remove the initial condition $y_0$ from the dataset. The horizontal line represents the ground truth.

## A.2 Chapter 4

### A.2.1 Robust losses

The generalised variational posterior obtained with some divergence $\mathcal{D}$ is given by :

$$q^*(\theta) = \arg\min_{q \in \mathcal{Q}} \mathcal{D}(q(\theta) \| p(\theta|x)). \tag{A.1}$$

For the two divergences $\mathcal{D} = \{D^{(\beta)}, D^{(\gamma)}\}$, the posterior can be rewritten with the optimization problem :

$$P(\ell_n, D, \mathcal{Q}) : q^*(\theta) = \arg\min_{q(\theta) \in \mathcal{Q}} \mathbb{E}_{q(\theta)}[\ell^{\mathcal{D}}(\theta, x)] + KLD(q(\theta)||p(\theta)), \tag{A.2}$$

where $\ell^{\mathcal{D}}$ corresponds to the divergence $\mathcal{D}$.

We denote $\ell^{\beta}$ and $\ell^{\gamma}$ the loss functions corresponding to the $\beta$ and $\gamma$ divergences respectively.

#### A.2.1.1 $\beta$-loss

The $\beta$ loss function, denoted as $\ell^\beta$, aims to minimise the $\beta$ divergence and links the parameter $\theta$ to the observations $x_{1:n}$ by:

$$\ell^\beta(\theta, x_{1:n}) = \sum_{i=1}^{n} \mathcal{L}_p^\beta(\theta, x_i) = \sum_{i=1}^{n} -\frac{1}{\beta - 1} p(x_i|\theta)^{\beta-1} + \frac{I_{p,\beta}(\theta)}{\beta}, \tag{A.3}$$

where $\beta \in \mathbb{R}\backslash\{0, 1\}$ with

$$I_{p,a}(\theta) = \int p(y|\theta)^a dy.$$

When we consider the framework in Chapter 3 described in the Algorithm 1, the observations $x_{1:n}$ and the ODE state solutions $y_{1:n}$ obtained with the parameter $\theta$ leads to the $\beta$ loss function :

$$\ell^\beta(\theta, x_{1:n}) = -\left( \sum_{i=1}^{n} \frac{1}{\beta} \frac{1}{\sqrt{2\pi\sigma^2}^\beta} \exp(-\frac{\beta}{2\sigma^2}(y_i - x_i)^2) - \frac{(2\pi\sigma^2)^{-\frac{\beta}{2}}}{(1+\beta)^{\frac{3}{2}}} \right). \tag{A.4}$$

*Proof.* With $\tilde{\beta} = \beta - 1$, we have

$$\ell^\beta(\theta, x_{1:n}) = \sum_{i=1}^{n} -\frac{1}{\tilde{\beta}} p(x_i|\theta)^{\tilde{\beta}} + \frac{I_{p,\tilde{\beta}+1}(\theta)}{\tilde{\beta}+1}, \tag{A.5}$$

$$= \sum_{i=1}^{n} -\frac{1}{\tilde{\beta}} (2\pi\sigma^2)^{-\frac{1}{2}\times\tilde{\beta}} \exp\left( -\frac{\tilde{\beta}}{2\sigma^2}(y_i - x_i)^2 \right) + \frac{1}{\tilde{\beta}+1} I_{p,\tilde{\beta}+1}(\theta). \tag{A.6}$$

The second part can be simplified into a constant when changing the noise from $\sigma^2$ to $\frac{\sigma^2}{\tilde{\beta}+1}$ (assuming $\tilde{\beta} > -1$).

$$I_{p,a}(\theta) = \int (2\pi\sigma^2)^{-\frac{1}{2}\times a} \exp\left( -\frac{1}{2\frac{\sigma^2}{\sqrt{a^2}}}(y_i - x_i)^2 \right) dx, \tag{A.7}$$

$$= (2\pi\sigma^2)^{-\frac{1}{2}\times a} \sqrt{2\pi(\frac{\sigma}{\sqrt{a}})^2} \int \frac{1}{\sqrt{2\pi(\frac{\sigma}{\sqrt{a}})^2}} \exp\left( -\frac{1}{2(\frac{\sigma}{\sqrt{a}})^2}(y_i - x_i)^2 \right) dx, \tag{A.8}$$

$$= \frac{(\sqrt{2\pi\sigma^2})^{1-a}}{\sqrt{a}}. \tag{A.9}$$

Therefore, we have:

$$\frac{1}{\tilde{\beta}+1} I_{p,\tilde{\beta}+1}(\theta) = \frac{1}{\tilde{\beta}+1} \frac{(\sqrt{2\pi\sigma^2})^{-\tilde{\beta}}}{\sqrt{\tilde{\beta}+1}} = \frac{(2\pi\sigma^2)^{-\frac{\tilde{\beta}}{2}}}{(\tilde{\beta}+1)^{\frac{3}{2}}}. \tag{A.10}$$

□

#### A.2.1.2 $\gamma$ loss

The $\gamma$ loss, represented as $\ell^\gamma$ and focused on minimizing the $\gamma$ divergence, can be expressed as follows:

$$\ell^\gamma(\theta, x_{1:n}) = \sum_{i=1}^{n} -\frac{1}{\gamma-1} p(x_i|\theta)^{\gamma-1} \times \frac{\gamma}{I_{p,\gamma}(\theta)^{\frac{\gamma-1}{\gamma}}}, \tag{A.11}$$

where $\gamma \in \mathbb{R}\backslash\{0,1\}$.

Similarly to the section above, we have the final expression for the loss given by :

$$\ell^\gamma(\theta, x_{1:n}) = -\sum_{i=1}^{n} \frac{1}{\gamma} \frac{1}{\sqrt{2\pi\sigma^{2\gamma}}} \exp(-\frac{\gamma}{2\sigma^2}(y_i - x_i)^2) \times \frac{\gamma+1}{\left[\frac{(\sqrt{2\pi\sigma^2})^{-\gamma}}{\sqrt{\gamma+1}}\right]^{\frac{\gamma}{\gamma+1}}}. \tag{A.12}$$

*Proof.* Again with $\tilde{\gamma} = \gamma - 1$.

$$\ell^\gamma(\theta, x_{1:n}) = -\frac{1}{\tilde{\gamma}} \frac{1}{\sqrt{2\pi\sigma^{2\tilde{\gamma}}}} \exp(-\frac{\tilde{\gamma}}{2\sigma^2}(y_i - x_i)^2) \times \frac{\tilde{\gamma}+1}{I_{p,\tilde{\gamma}+1}(\theta)^{\frac{\tilde{\gamma}}{\tilde{\gamma}+1}}} \tag{A.13}$$

It follows from the previous section that

$$I_{p,\tilde{\gamma}+1}(\theta)^{\frac{\tilde{\gamma}}{\tilde{\gamma}+1}} = \left[\frac{(\sqrt{2\pi\sigma^2})^{-\tilde{\gamma}}}{\sqrt{\tilde{\gamma}+1}}\right]^{\frac{\tilde{\gamma}}{\tilde{\gamma}+1}}. \tag{A.14}$$

□

## A.3 Chapter 5

| dGp A | $\delta$ | 0.0 | 0.05 | 0.1 | 0.2 |
|---|---|---|---|---|---|
| D = 0 | n=20 | 10.009;0.001 | 8.737;0.002 | 7.718;0.002 | 6.199;0.002 |
| | n=50 | 10.006;0.003 | 8.763;0.001 | 7.749;0.002 | 6.221;0.002 |
| | n=100 | 10.028;0.004 | 8.756;0.001 | 7.746;0.002 | 6.247;0.002 |
| | n=150 | 10.015;0.002 | 8.768;0.002 | 7.762;0.002 | 6.22;0.003 |
| | n=200 | 10.010;0.001 | 8.766;0.002 | 7.767;0.002 | 6.253;0.002 |
| D = 5 | n=20 | 9.55;0.003 | 8.044;0.002 | 6.322;0.002 | 4.282;0.002 |
| | n=50 | 9.527;0.002 | 8.061;0.003 | 6.193;0.002 | 4.686;0.002 |
| | n=100 | 9.541;0.002 | 8.086;0.003 | 6.592;0.004 | 4.663;0.003 |
| | n=150 | 9.543;0.001 | 8.112;0.004 | 6.582;0.004 | 4.654;0.002 |
| | n=200 | 9.581;0.0005 | 8.02;0.001 | 6.03;0.002 | 4.049;0.002 |
| D = 10 | n=20 | 10.026;0.003 | 8.155;0.004 | 6.492;0.002 | 4.172;0.003 |
| | n=50 | 10.038;0.004 | 8.116;0.005 | 6.469;0.002 | 3.762;0.002 |
| | n=100 | 9.908;0.005 | 8.16;0.002 | 6.45;0.002 | 3.812;0.002 |
| | n=150 | 10.173;0.002 | 8.127;0.001 | 6.268;0.002 | 4.424;0.002 |
| | n=200 | 9.811;0.003 | 8.337;0.005 | 6.593;0.002 | 4.559;0.003 |
| D = 15 | n=20 | 10.691;0.003 | 8.715;0.004 | 7.344;0.013 | 3.853;0.003 |
| | n=50 | 9.881;0.01 | NA | NA | NA |
| D = 20 | n=20 | 10.255;0.013 | 8.688;0.002 | 6.697;0.003 | 4.233;0.005 |
| | n=50 | 10.066;0.006 | 8.137;0.008 | 8.778;0.005 | 3.838;0.006 |
| | n=100 | 10.293;0.006 | 8.77;0.003 | 7.565;0.01 | 3.999;0.005 |
| | n=150 | 10.112;0.001 | 8.707;0.005 | 6.129;0.005 | 4.641;0.005 |
| | n=200 | 9.231;0.006 | 8.836;0.003 | 7.788;0.002 | 3.913;0.003 |
| D = 50 | n=20 | 10.27;0.016 | 8.91;0.002 | 7.897;0.071 | 4.425;0.04 |
| | n=50 | 9.496;0.036 | NA | NA | NA |
| | n=100 | 9.917;0.002 | NA | NA | NA |
| | n=150 | 9.988;0.018 | NA | NA | NA |
| D = 100 | n=20 | 10.026;0.006 | 8.169;0.022 | 8.18;0.009 | 6.82;0.005 |
| | n=50 | 10.523;0.015 | NA | NA | NA |
| | n=100 | 9.445;0.017 | NA | NA | NA |
| | n=150 | 10.361;0.107 | NA | NA | NA |
| D = 200 | n=20 | 9.461;0.007 | 8.664;0.096 | 7.937;0.025 | 9.327;0.01 |

Table A.3: Variational Posterior Analysis for Model A1 using observations from the dGp A with RFF-VAE denoted $\mu_\phi; \sigma^2_\phi$ across varying data size $n$, misspecification error $\delta$, and RFF size $D$. NA denotes that the computation has not been performed.

| dGp B | $\delta$ | 0.0 | 0.05 | 0.1 | 0.2 |
|---|---|---|---|---|---|
| D = 0 | n=20 | 9.985;0.001 | 9.974;0.0 | 9.993;0.0 | 10.065;0.0 |
| | n=50 | 9.973;0.001 | 9.976;0.002 | 9.994;0.002 | 10.001;0.002 |
| | n=100 | 9.995;0.001 | 9.969;0.004 | 9.966;0.002 | 9.962;0.003 |
| | n=150 | 9.967;0.002 | 9.991;0.001 | 9.966;0.003 | 9.977;0.002 |
| | n=200 | 9.966;0.003 | 9.923,0.003 | 9.961;0.005 | 9.976;0.001 |
| D = 5 | n=20 | 8.887;0.0 | 11.486;1.79 | 9.009;0.0 | 8.84;0.015 |
| | n=50 | 9.417;0.001 | 9.443;0.002 | 9.749;0.002 | 9.76;0.002 |
| | n=100 | 9.482.0.002 | 9.527;0.002 | 9.722;0.003 | 9.883;0.007 |
| | n=150 | 9.626;0.001 | 9.695;0.001 | 9.819;0.003 | 9.915;0.006 |
| | n=200 | 9.138;0.001 | 9.614;0.002 | 10.053;0.005 | 9.606;0.003 |
| D = 10 | n=20 | 8.722;0.002 | 9.196;0 | 9.032;0 | 9.629;0.001 |
| | n=50 | 9.29;0.003 | 9.106;0.016 | 7.921;0.043 | 9.18;0.033 |
| | n=100 | 8.188;0.045 | 8.903;0.094 | 8.658;0.04 | 9.251;0.022 |
| | n=150 | 9.314;0.006 | 8.933;0.01 | 8.803;0.003 | 9.119;0.084 |
| | n=200 | 9.299;0.012 | 9.351;0.01 | 9.454;0.002 | 9.194;0.017 |
| D = 15 | n=20 | 8.832;0.002 | 8.719;0 | 8.818;0.001 | 9.011;0.001 |
| | n=50 | 9.369;0 | 9.405;0.001 | 9.415;0 | 9.267;0.002 |
| D = 20 | n=20 | 9.252;0.001 | 13.631;0.053 | 9.574;0.011 | 17.28;0 |
| | n=50 | 9.915;0.079 | 8.077;0.078 | 9.762;1.091 | 9.18;0.033 |
| | n=100 | 8.693;0.011 | 8.586;0.046 | 9.53;0.078 | 8.762;0.082 |
| | n=150 | 9.546;0.06 | 9.324;0.065 | 9.335;0.067 | 9.502;0.057 |
| | n=200 | 8.8;0.022 | 9.765;2.113 | 9.22;0.061 | 9.745;0.069 |
| D = 50 | n=20 | 11.747;0.153 | 11.101;0.06 | 12.444;0.135 | 11.0;0.757 |
| | n=50 | 8.796;4.133 | 10.18;3.573 | 8.849;2.274 | 11.542;0.66 |
| | n=100 | 10.776;0.059 | 9.766;0.05 | 9.615;0.083 | 11.269;5.348 |
| | n=150 | 10.682;0.691 | 7.783;5.525 | 10.435;0.047 | 9.846;0.066 |
| | n=200 | 10.039;0.053 | 10.668;0.398 | 9.611;0.081 | 7.069;0.069 |
| D = 100 | n=20 | 9.463;0.04 | 10.54;0.026 | 10.279;0.083 | 9.078;0.285 |
| | n=50 | 10.095;0.088 | 10.466;0.047 | 12.873;0.044 | 9.453;4.123 |
| | n=100 | 9.372;0.1 | 9.765;4.868 | 10.018;5.538 | 8.114;0.078 |
| | n=150 | 9.917;0.069 | 8.626;0.065 | 9.359;1.704 | 9.53;5.529 |
| | n=200 | 8.723;1.396 | 9.548;5.503 | 9.809;1.689 | 9.455;0.083 |
| D = 200 | n=20 | 12.139;0.017 | 9.087;0.02 | 11.167;0.182 | 7.76;0.025 |
| | n=50 | 8.412;0.033 | 16.694;0 | 19.669;0.543 | 9.528;0.09 |

Table A.4: Variational Posterior Analysis for Model B1 using observations from the dGp B with RFF-VAE denoted $\mu_\phi; \sigma^2_\phi$ across varying data size $n$, misspecification error $\delta$, and RFF size $D$.

| dGp C | $\delta$ | 0.0 | 0.05 | 0.1 | 0.2 |
|---|---|---|---|---|---|
| D = 0 | n=20 | 9.226;0.014 | 9.22;1.579 | 9.207;1.97 | 9.985;0.014 |
| | n=50 | 9.24;0.014 | 9.231;1.547 | 9.943;1.579 | 9.955;1.579 |
| | n=100 | 8.418;0.016 | 9.885;2.162 | 9.89;2.176 | 9.9;0.019 |
| | n=150 | 9.85;2.763 | 9.87;0.024 | 8.65;0.024 | 8.656;2.763 |
| | n=200 | 8.88;0.001 | 9.768;0.046 | 9.715;0.046 | 9.713;0.046 |
| D = 5 | n=20 | 8.654;0.246 | 9.133;0.031 | 8.542;0.128 | 9.032;0.351 |
| | n=50 | 9.557;1.583 | 9.427;0.119 | 9.05;1.53 | 9.953;1.592 |
| | n=100 | 9.617;2.247 | 9.567;0.02 | 8;746;0.712 | 9.871;0.024 |
| | n=150 | 9.556;2.763 | 8.398;0.024 | 8.584;0.03 | 8.463;0.367 |
| | n=200 | 8.464;0.501 | 9.582;5.524 | 7.469;1.834 | 7.311;5.327 |
| D = 10 | n=20 | 8.974;0.045 | 8.407;9.961 | 8.689;2.211 | 9.24;1.313 |
| | n=50 | 9.233;1.003 | 9.199.1.9 | 8.439;1.851 | 8.405;2.607 |
| | n=100 | 8.842;1.033 | 8.733;3.454 | 8.056;0.076 | 9.393;1.896 |
| | n=150 | 8.933;0.735 | 7.366;2974 | 9.575;1.382 | 10.16;3.416 |
| | n=200 | 7.787;0.773 | 9.876;5.474 | 9.201;0.145 | 6.971;5.568 |
| D = 15 | n=20 | 9.217;0.049 | 8.267;1.044 | 10.344;0.717 | 9.059;0.037 |
| | n=50 | 8.277;0.203 | 9.834;0.042 | 14.446;6.942 | 9.692;0.125 |
| D = 20 | n=20 | 10.233;0.493 | 9.749;1.891 | 8.087;0.064 | 8.577;6.396 |
| | n=50 | 9.983;5.669 | 9.218;22.052 | 10.767;0.1 | 12.456;189.447 |
| | n=100 | 9.543;2.44 | 10.836;2.382 | 9.512;0.117 | 8.324;1.327 |
| | n=150 | 9.042;15.207 | 9.084;0.307 | 9.661;0.314 | 8.627;3.071 |
| | n=200 | 7.204;0.339 | 9.573;0.086 | 11.427;111.699 | 7.206;5.278 |
| D = 50 | n=20 | 4.595;1.183 | 3.772;11.949 | 4.151;0.218 | 8.047;9.327 |
| | n=50 | 13.398;65.254 | 14.05;7.727 | 11.964;4.694 | 15.476;19.854 |
| | n=100 | 8.137;197.636 | 9.582;0.35 | 9.985;0.481 | 10.107;31.697 |
| | n=150 | 15.223;0.014 | 10.553;35.989 | 14.648;5.53 | 9.998;1.73 |
| | n=200 | 9.563;7.224 | 11.38;2.812 | 10.154;20.094 | 10.307;17.182 |
| D = 100 | n=20 | 3.68;0.146 | 4.364;0209 | 7.434;5.462 | 5.092;2.909 |
| | n=50 | 9.303;64.413 | 10.293;62.655 | 7.782;127.826 | 7.515;25.178 |
| | n=100 | 9.809;4.437 | 11.983;4372.309 | 8.161;2.046 | 12.332;1.165 |
| | n=150 | 9.211;0.115 | 3.887;10.58 | 13.005;0.254 | 10.609;13.022 |
| | n=200 | 14.826;50.994 | 12.33;0.046 | 10.281;0.642 | 7.089;2.328 |
| D = 200 | n=20 | 0.277;0.385 | 3.845;0.001 | 6.777;665.109 | 0.871;0.131 |
| | n=50 | 8.92;0.001 | 4.155;3.027 | 13.884;0.006 | 6.286;0.15 |

Table A.5: Variational Posterior Analysis for Model C1 using observations from the dGp C with RFF-VAE denoted $\mu_\phi; \sigma^2_\phi$ across varying data size $n$, misspecification error $\delta$, and RFF size $D$.

# Appendix B

# Additional details

**Theorem B.1** (GVI modularity)**.**

*For Bayesian inference with $P(\ell_n, D, \Pi)$, making it robust to model misspecification amounts to changing $\ell_n$. Conversely, adapting uncertainty quantification amounts to changing $D$.*

**Proposition 3.** *The Kullback–Leibler divergence between two multivariate Gaussian distributions $\mathcal{N}(\boldsymbol{\mu_p}, \Sigma_p)$ and $\mathcal{N}(\boldsymbol{\mu_q}, \Sigma_q)$, both $k$ dimensional is :*

$$D_{KL}(p||q) = \frac{1}{2}\left[\log\frac{|\Sigma_q|}{|\Sigma_p|} - k + (\boldsymbol{\mu_p} - \boldsymbol{\mu_q})^T\Sigma_q^{-1}(\boldsymbol{\mu_p} - \boldsymbol{\mu_q}) + tr\left\{\Sigma_q^{-1}\Sigma_p\right\}\right].$$

*When $q$ is $\mathcal{N}(0, I)$, we get,*

$$D_{KL}(p||q) = \frac{1}{2}\left[\boldsymbol{\mu_p}^T\boldsymbol{\mu_p} + tr\left\{\Sigma_p\right\} - k - \log|\Sigma_p|\right].$$